Proficy®
Logic Developer®

GETTING STARTED

Version 8.00
2013
GFK-1918R
All rights reserved. No part of this publication may be reproduced in any form or by any electronic or mechanical means, including photocopying and recording, without permission in writing from GE Intelligent Platforms, Inc.

Disclaimer of Warranties and Liability

The information contained in this manual is believed to be accurate and reliable. However, GE Intelligent Platforms, Inc. assumes no responsibilities for any errors, omissions or inaccuracies whatsoever. Without limiting the foregoing, GE Intelligent Platforms, Inc. disclaims any and all warranties, expressed or implied, including the warranty of merchantability and fitness for a particular purpose, with respect to the information contained in this manual and the equipment or software described herein. The entire risk as to the quality and performance of such information, equipment and software, is upon the buyer or user. GE Intelligent Platforms, Inc. shall not be liable for any damages, including special or consequential damages, arising out of the use of such information, equipment and software, even if GE Intelligent Platforms, Inc. has been advised in advance of the possibility of such damages. The use of the information contained in the manual and the software described herein is subject to GE Intelligent Platforms, Inc. standard license agreement, which must be executed by the buyer or user before the use of such information, equipment or software.

Notice

GE Intelligent Platforms, Inc. reserves the right to make improvements to the products described in this publication at any time and without notice.

© 2013 GE Intelligent Platforms, Inc. All rights reserved. * Trademark of GE Intelligent Platforms, Inc. Any other trademarks referenced herein are used solely for purposes of identifying compatibility with the products of GE Intelligent Platforms, Inc.

We want to hear from you. If you have any comments, questions, or suggestions about our documentation, please send them to the following e-mail address:

doc.ip@ge.com
Contents

1 Welcome 5
   System Requirements ........................................... 6
   Installation .................................................... 7
   Product Authorization ........................................... 8
      Hardware Key Authorization ................................. 8
      Software Key ................................................. 9
   Contact information ........................................... 13
      General Contact Information ............................... 13
      Technical Support .......................................... 13
      Americas .................................................... 13
      Europe, Middle East, and Africa (EMEA) ..................... 13
      Asia Pacific ............................................... 14

2 Proficy* Machine Edition* 15
   Quick Start ................................................... 16
   Projects ....................................................... 18
   Tools ........................................................... 20
      Using docking markers .................................... 21
   Variables ....................................................... 22
      User-defined Data Types .................................. 23
      Variable properties in the Inspector .................... 23
   Options ........................................................ 26
   Machine Edition Help ........................................ 27
   Companion Help ............................................... 27
   InfoViewer Help ................................................ 27

3 GE IP Targets 33
   Adding, Configuring, and Converting Targets ................ 34
   Adding Targets .................................................. 34
   Configuring Controller Targets ................................ 35
Converting Targets ............................................. 36
Configuring Communication .................................... 38
Interacting with a Controller ................................ 44
Validating a Target ............................................ 44
Offline, Online: monitor mode, programmer mode .......... 44
Upload/Download ............................................. 47
Run/Stop ...................................................... 49
Fault Tables .................................................. 50
Reference View Tables ........................................ 52
Reports .......................................................... 53

4 Hardware Configuration ........................................ 55
PACSystems* RXi .............................................. 55
PACSystems* RX7i ............................................. 57
  Configuring PACSystems* RX7i Hardware ................. 58
PACSystems* RX3i ............................................. 59
  Configuring PACSystems* RX3i Hardware ................. 60
Configuring Controller Hardware ................................ 61
I/O variables .................................................. 67
Hot Redundancy Systems ..................................... 70
  Configuring Hot Redundancy Systems ...................... 72
DSM324i and Motion Mate DSM314 Motion Modules .......... 76
Remote I/O ................................................... 78
  Series 90*-70 Genius Remote I/O Scanner ................. 78
  VersaMax* Remote I/O ..................................... 80

5 Logic Programs and Blocks .................................. 85
Program Types ................................................ 85
Number of Blocks in the Main Program ...................... 88
Scheduling Programs ......................................... 88
Indirect References .......................................... 91
LD Editor ..................................................... 93
  Working with the LD Editor Offline .................... 94
  Editing Logic as Text ................................... 97
  Working with the LD Editor Online .................... 99
Test Edit (PACSystems* only) ................................ 100
  Word-for-Word Changes ................................. 102
10 Motion Programming 153

Motion Editor ................................................................. 154
Working with the Motion Editor ................................. 155
Motion Commands ................................................................. 156
Local Logic ................................................................. 157
  Working with the Local Logic Editor ................. 158
  Local Logic Variables ........................................ 159
  Local Logic Commands and Operators .......... 160
CAM Editor ................................................................. 161
  Working with the Cam Editor ...................... 162

Index 165
Congratulations on your purchase of Proficy® Logic Developer - PLC®, the PACSystems®, Series 90®, and VersaMax® Controller programming component of Proficy® Machine Edition® automation software.

This software package provides all the tools necessary to create powerful control applications. Logic Developer - PLC provides a way to configure your Controller hardware or remote I/O, create and edit logic, upload and download projects, and monitor and debug the execution of control programs. Projects can be imported from Logicmaster, VersaPro, and Control folders.

Hosted in the Machine Edition environment, Logic Developer - PLC takes advantage of a powerful set of common programming tools (see page 20). The same tools can be applied to Logic Developer - PC® (PC Control), Logic Developer - State®, View, and Motion components, providing a single programming environment. The Machine Edition environment unites and organizes components, providing data sharing and networked operation.

The following features are included in this version of Logic Developer - PLC:

- Hardware Configuration. See page 55.
- LD Editor. See page 93.
- FBD Editor. See page 108.
- IL Editor. See page 120.
- ST Editor. See page 127.
- C Blocks. See page 135.
- C Programs. See page 137.
- DLBs. See page 139.
- PACMotion. See page 149.
- Motion Editor. See page 153.
- Local Logic Editor. See page 157.
- CAM Editor. See page 161.
SYSTEM REQUIREMENTS

To use Logic Developer - PLC* and its tools, you require the following:

- 32-bit or 64-bit Windows 7 SP1 Ultimate, Windows 7 SP1 Enterprise, Windows 7 SP1 Professional; 32-bit Windows XP Professional SP2 or SP3.
  - In Windows 7 SP1, you must be part of the Administrators group. In Windows XP, you must be part of the PowerUsers group or the Administrators group.
  - Windows regional settings must be set to English.
  - Windows XP Tablet PC Edition and Windows XP Embedded are not supported.
  - The following features are not supported in 64-bit Windows platforms:
    - The property column display in the Variables tab of the Navigator; EGD Configuration Server and EGD Management Tool (workaround: configure EGD manually).
- Internet Explorer 9.0, 8.0, 7.0, 6.0 SP1 or later, or 5.5 with SP2 Q810847 or later.
  - You must install Internet Explorer before installing Machine Edition*.
- TCP/IP network protocol (if you use an Ethernet connection).
- Requirements for processor speed and memory:
  - Minimum: Intel Core Duo or AMD Athlon 64 X2 with 4 GB RAM.
  - Recommended: 2.6 GHz Intel Core i series with 4 GB RAM.
- .NET Framework 4.0 Full. If the Microsoft .NET Framework is not yet installed, it is automatically included during the installation of Machine Edition, and a reboot may be required to complete the installation.
- 2 GB hard disk space. Additional space is required for your projects and temporary files.

The Development Environment is supported on VMware Player 2.0 or greater or VMware Workstation 6.0 or greater

Note: The APM Motion Programmer does not function if Machine Edition runs within VMware.
INSTALLATION

For last-minute information, release notes, and supported hardware lists for Machine Edition* products, see the Important Product Information (IPI) document on the install disk. There are several ways to view this document:


If you have a previous version of Machine Edition installed on your computer, you must uninstall it before installing the latest version. All of your existing projects, settings, and authorizations are preserved following an uninstall operation.

To install Logic Developer - PLC*

1. Insert the Machine Edition install disk into your CD-ROM drive.
   By default, the setup program automatically starts. If the setup program does not automatically start, run Setup.exe in the root directory of the install disk.
2. Click Install to start the install process.
3. Follow the instructions as they appear on the screen.
   For information on troubleshooting installation problems, see the IPI Release Notes.
PRODUCT AUTHORIZATION

A new installation of Logic Developer - PLC provides a 4-day trial license with full access to Logic Developer - PLC features. This license overrides all other licensing and cannot be removed. Any licensing added is apparent on the fifth day.

To continue using Logic Developer - PLC beyond the 4-day trial period, you must authorize the software.

Two types of authorization are available: software key authorization and hardware key authorization. On a single computer, it is not recommended to use both types of authorization.

Hardware Key Authorization

A M4 or MAX USB hardware key contains authorization files that are read by the Proficy Common License Viewer, which enables Proficy products, including Logic Developer - PLC, to execute.

To see what products and target types are authorized by a hardware key

- On the operating system Start menu, point to Programs, then to Proficy Common, and then choose License Viewer.

To authorize Logic Developer - PLC with a hardware key

1. Find out which version of the Proficy Licensing Software is installed on your computer: On the operating system's Start menu, point to Programs, then to Proficy Common, and then choose License Viewer. In the top right corner of the License Viewer window that appears, click About. Take note of the version number that appears and click OK.

2. If this is the first time you install a Proficy product, access http://support.ge-ip.com, and in the top right corner, click Register. In the Single Sign On page that appears, enter your email address and click Submit. When you receive a GE Intelligent Platforms Web Access Registration email, follow the instructions. In the Single Sign On -- SSO Registration application window that appears, fill out the information and click Submit.

3. Access http://support.ge-ip.com/support/index?page=securitykey&id=ST15&lclicked=Licensing Key Updates (login req). Enter your SSO User ID and Password. In the middle of the page that appears, follow the Updating the Max/M4 or Pro/M1 key procedure, keeping in mind the following:
Logic Developer - PLC does not support the insertion of a hardware key into a parallel port, the insertion of hardware keys in multiple USB ports, or the combination of a hardware key with a software key on the same computer.

If the web site refers to a more recent version than the one on your computer, uninstall your current version, download the more recent version, and install it.

The USB port is usually located on the front or back of your tower case, or in the side of a laptop. We do not recommend using a USB port on your monitor.

If this is a new hardware key, double-click the license file <serialnumber>.plic (supplied by your local distributor). In the wizard that appears, accept all the defaults. The license file will be flashed onto your key.

The first time you plug the key into a USB port, you see a progress dialog box that indicates that Windows has detected new hardware and is updating its system settings. After the dialog box disappears, the settings stored in the hardware key are active and you can use the products that it authorizes.

You must leave the hardware key in the port while using the products. If you remove the key from the port, authorization for those products is removed and replaced with read-only access to your projects.

To move the hardware-key authorization to another computer

- Remove the hardware key from the source computer and follow the previous procedure on the destination computer.

After the authorization file <serialnumber>.plic has been flashed to the M4 or MAX hardware key, it can be used on any computer on which the current version of the Proficy Common License Viewer is installed.

Software Key

When authorizing Logic Developer - PLC with a software key, you need to contact us by telephone, fax, or e-mail. Authorization personnel are available on the telephone between 6 a.m. and 6 p.m. MST.

Software key authorization is specific to a single computer. If you want to work with Logic Developer - PLC on a different computer, you must move the authorization to that second computer (see page 10).

To see what products and target types are authorized by a software key

- On the operating system Start menu, point to Programs, then to Proficy (or GE Fanuc or CIMPLICITY Machine Edition, depending on when Machine Edition
Welcome

Product Authorization

was first installed on the computer), then to Proficy Machine Edition, and then choose Product Authorization.

To authorize a copy of Logic Developer - PLC with a software key

1. Have your serial number ready. The serial number can be found on the Certificate of Authentication that came with Logic Developer - PLC.

2. On the operating system's Start menu, point to Programs, then to Proficy (or GE Fanuc or CIMPLICITY Machine Edition, depending on when Machine Edition was first installed on the computer), then to Proficy Machine Edition, and then choose Product Authorization.

3. In the Product Authorization dialog box that appears, click Add.

4. Select how to authorize: by Internet, Phone/Fax/E-mail, or Floppy Disk Transfer. Click Next.

   If you choose the Internet option, follow the instructions on the web site.

   If you choose the Phone/Fax/E-mail option, proceed to step 5.

   If you choose the Floppy Disk Transfer option, ensure that you have an authorization removable media before proceeding.

5. Fill in the fields in the dialog box. Fields that are identified with an asterisk (*) must be filled in.

6. If authorizing by:

   - **Phone, click Phone/Fax** and call the number on the screen to receive one or more new key codes.

   - **Fax**, click Phone/Fax. In the dialog box that appears, click Print FAX. Fax the Product Authorization Request to us, using our fax number on the printout. We will reply by fax with your new key code(s).

   - **Internet**, On the Authorization web page, click Submit Authorization. We will then reply by e-mail with your new key code(s).

   - **E-mail**, click Send E-mail. In the dialog box that appears, click Authorize to e-mail us. We will reply by e-mail with your new key code(s).

Product Authorization is complete after you enter the new key code and it has been accepted. If you have purchased other Proficy products, you may need to run the Product Authorization program for each product.

To move the software-key authorization to another computer

You can run the software only on the computer that has Product Authorization. To develop your projects on a different computer, you need to complete the
following steps to move the software key authorization from one computer to another. (For a more detailed procedure with screen shots, access www.ge-ip.com/support and look up Knowledge Base article KB5659.)

On the computer that the authorization will be moved to, do as follows:

1. Install Logic Developer - PLC*
2. On the operating system’s Start menu, point to Programs, then to Proficy (or GE Fanuc or CIMPLICITY Machine Edition, depending on when Machine Edition was first installed on the computer), then to Proficy Machine Edition, and then choose Product Authorization.
3. In the top right corner of the Product Authorization dialog box that appears, take note of the Site Code. This code is required to generate the license file on the source computer.
4. Leave the dialog box open.

On the source computer, do as follows:

1. On the operating system’s Start menu, point to Programs, then to Proficy (or GE Fanuc or CIMPLICITY Machine Edition, depending on when Machine Edition was first installed on the computer), then to Proficy Machine Edition, and then choose Product Authorization.
2. In the Product Authorization dialog box that appears, select Logic Developer - PLC and click Move.
3. In the Before Moving message box that appears, read the note and click OK.
4. Enter the Site Code that you wrote down when you were on the destination computer.
   Caution: If the Site Code is entered incorrectly, the authorization will be lost and cannot be recovered; you will need to contact technical support (see page 13).
5. Click Next.
6. In the Confirm dialog box that appears, verify that the site code is correct. If it is, click OK.
7. Insert a blank formatted removable medium in the appropriate location.
8. In the refreshed Product Authorization dialog box, select the drive letter mapped to the medium you inserted.
   Do not select the C:\ drive; otherwise, your source license files will be overwritten.
9. Click Next.
10. In the window that appears, confirm the accuracy of the Target Site Code. If it is correct, click Finish.
Welcome
Product Authorization

The authorization code is moved to a file named authoriz.dat on the removable medium you selected.

11. In the Authorization disk successfully created message box, read the message and click OK.

On the destination computer, do as follows:

1. Insert the removable medium.
2. In the Product Authorization window that you left open earlier, click Add.
3. In the refreshed Product Authorization dialog box, select Floppy Disk Transfer and click Next.
4. In the refreshed dialog box, to the right of the Please insert your Authorization Disk field, click the button.
5. In the dialog box that appears, select the drive letter that corresponds to the medium you inserted and navigate to the location of the authoriz.dat file. Click Open.
6. In the Product Authorization dialog box, click Next.
7. In the refreshed dialog box, verify the product and contact information. If all is correct, click Finish.
8. In the message box that indicates that the move was successful, click OK.

The authorization has now been moved to the new computer, as you can see in the refreshed Product Authorization window.

9. Click Exit.

To change a software key to a hardware key or vice-versa

• Contact your local distributor.
CONTACT INFORMATION

If you purchased this product through an Authorized Channel Partner, please contact them directly.

General Contact Information

Online Technical Support and GlobalCare: www.ge-ip.com/support
Comments about our manuals or online help: doc.ip@ge.com
Additional information: www.ge-ip.com
Solution Provider: solutionprovider.ip@ge.com
Authorization: authorization.ip@ge.com

Technical Support

If you have technical problems that cannot be resolved with the information in this guide, look up the Machine Edition online help. If you still need help, please contact us by telephone or email, or on the web at www.ge-ip.com/support.

Americas

Online Technical Support: www.ge-ip.com/support
Phone: 1-800-433-2682
International Americas direct dial: 1-434-978-5100 (if toll free 800 option is unavailable)
Technical Support email: support.ip@ge.com
Customer Care email: customercare.ip@ge.com
Inside Sales: insidesales.ip@ge.com
Primary language of support: English

Europe, Middle East, and Africa (EMEA)

Online Technical Support: www.ge-ip.com/support
Phone: +800 1-433-2682
EMEA direct dial: +1 780-401-7717 if the toll free 800 option is unavailable or if you are dialing from a mobile telephone
Technical Support email: support.emea.ip@ge.com
Customer Care email: customercare.emea.ip@ge.com
Inside Sales: insidesales.emea.ip@ge.com
Welcome
Contact information

Primary languages of support: English, French, German, Italian, Czech, Spanish

Asia Pacific
Online Technical Support and GlobalCare: www.ge-ip.com/support
Phone: +86-400-820-8208
+86-21-3217-4826 (India, Indonesia, and Pakistan)
Technical Support email: support.cn.ip@ge.com (China)
support.jp.ip@ge.com (Japan)
support.in.ip@ge.com (other Asian locales)
Customer Care email: customercare.apo.ip@ge.com
customercare.cn.ip@ge.com (China)
Proficy Machine Edition offers you a complete solution for the development of automation applications, in one package. Machine Edition features an integrated development environment and tools that enable you to spend more time building applications and less time learning the software. All Machine Edition products are fully integrated into the environment and interact with each other.

- They share the same set of tools providing a consistent interface throughout the development process.
- Full drag-and-drop capabilities between tools and editors.
- A true scalable solution: you choose the type of Controller your projects run on.

The following illustrates a Machine Edition project:
QUICK START

Machine Edition* makes it easy to get started developing a project.

To start Machine Edition

1. Click , point to Programs, point to Proficy, point to Proficy Machine Edition, and then choose Proficy Machine Edition.

After Machine Edition initializes, the Environment Themes dialog box appears.

Note: The Environment Themes dialog box appears automatically the first time you start Machine Edition. To change the theme later, choose Apply Theme from the Windows menu.

2. Choose the Logic Developer - PLC theme.

3. Click OK.

When you open a project, the appearance of your Machine Edition screen matches the preview displayed in the Environment Themes dialog box.

Select this option if you want to create a new project from scratch.

Select this option to use a template to create your project.

Select this option if you want to choose an existing project.

Select this option to display only the projects that have been used recently.

Select this option to display all projects stored on your computer.

4. Select the appropriate option to open a project. The Open an existing project option is selected by default.

Notes

- If you select either the Empty project option or the Machine Edition template option, the New Project dialog box appears and you can continue creating a new project (see page 18).
- If you select the Open an existing project option, you can also select either the Recent Projects option or the All Projects option. The Recent Projects option is selected by default.

5. If you selected the Open an existing project option, select from the list the project that you want to open.

6. If you want, select the Don’t show this dialog box on startup option.

7. Click OK.

PROJECTS

You can create and edit Machine Edition projects by using products such as View, Logic Developer - PC, Motion, Logic Developer - State, and Logic Developer - PLC. These products share Machine Edition tools to provide a high level of integration between the various parts of your project.

You can import folders created with Logicmaster, Control, or VersaPro.

With Logic Developer - PLC, you can build multiple projects to suit your specific requirements.

To create a new project using a template

Before creating a project, there are some things you should know:

- The primary components your project will contain.
- The GE IP Controller your project will run on.

1. From the File menu, choose New Project, or click on the File toolbar.

The New Project dialog box appears.

2. Choose a Project Template that suits your needs.

3. Enter a descriptive Project Name.

4. Click OK.

Your project opens in the Machine Edition environment.
To open an existing project for editing

1. Open the Navigator and select the Manager tab.
   A list of projects appears.

2. Do one of the following:
   - Right-click a project and choose Open.
   - or -
   - Double-click a project.

Your project loads and is ready for editing.

Want to know more? In the online help index, look up “Projects”.

To import a folder

1. Open the Navigator and select the Project tab.
2. Select the target that you want to import the folder into.
3. Right-click the target, point to Import, and choose the folder type.
4. In the dialog box that appears, navigate to and double-click the folder you want to import.

Want to know more? In the online help index, look up “importing Control folders”, “importing Logicmaster folders”, or “importing VersaPro folders”.

Navigator: Manager tab
TOOLS

Project development is supported by the Machine Edition* tools. Each tool is opened by means of a button on the Tools toolbar. When open, each tool is a docking window.

A description of each tool's function is provided below.

Navigator

The Navigator contains a set of tabs. Each tab displays information about your development system in a hierarchical tree structure similar to Windows Explorer. The available tabs depend on which Machine Edition products you have installed and what kind of work you are developing or managing. The Project tab shows the overall organization of your application.

Feedback Zone

The Feedback Zone window is used to display several types of output information generated by Machine Edition components. This interactive window uses category tabs to organize the output generated from the Machine Edition products you have installed.

Inspector

The Inspector window lists the properties and current settings for a selected object or element. You can edit these properties directly in the Inspector. When you select several objects, the Inspector window lists the properties common to all of them. The Inspector provides a simple method of viewing and setting properties for all Machine Edition objects.

Data Watch

The Data Watch tool is a run-time debugging tool that enables you to monitor and edit the values of variables. This tool is useful while working online to a target. With the Data Watch tool, you can monitor individual variables or user-defined lists of variables. Data Watch lists can be imported, exported, or saved with a project.

Toolchest

The Toolchest is a powerful storehouse of objects you can add to your project. You can drag most items directly from the Toolchest into Machine Edition editors. You can choose from predefined objects or create your own reusable fxClasses. The Toolchest adds true object-oriented capability to Machine Edition.
The Companion provides useful tips and information while you work. While the Companion is open, it tracks your moves and displays help on whatever item is currently selected in the Machine Edition* environment. It is context-sensitive and displays a description of whatever you click in Machine Edition.

The InfoViewer is an embedded Web browser used mainly to display the following:

- Machine Edition help
- Machine Edition reports
- The documentation associated with a project or target

If you are familiar with Internet Explorer or Netscape Navigator, then you are already familiar with the basic InfoViewer interface. Like the Companion, the InfoViewer is context-sensitive. Press F1 to get help on any item you select in the Machine Edition environment.

A table of contents is found in the InfoView tab of the Navigator.

Want to know more? In the online help index, look up "Tools: an Overview".

Using docking markers

If the Show Docking Markers option is enabled, you can use docking markers to help dock a tool window to a desired location.

As you drag a dockable window, a series of docking markers appear, indicating valid docking locations. Docking markers appear as a series of blue arrows. Move the mouse over a docking marker to preview how the window would be docked when using that marker. Release the mouse button while hovering over a marker to use that location.

These markers dock tools to the edge of an existing tool or to the Machine Edition Environment window.

This marker "stacks" the tool overtop an existing tool window, forming a series of tab-docked windows. You can then switch between tab-docked tools by clicking the tabs that appear at the bottom of the tab-docked windows.
Variables

A variable is a named storage space for data. That is, the variable name represents a memory location in the target Controller. The way in which the variable represents a memory location is determined by the value of the variable’s Ref Address property. A variable’s Ref Address property can be set to any of the following:

- **A reference address**, for example, %R00001. This is the GE IP Controller memory location that contains the variable’s value. This can be an I/O register or an internal memory register on the Controller.

- **A blank**. On a PACSystems*, if you leave a variable’s Ref Address property blank, the variable is a symbolic variable. Machine Edition* handles the mapping for you in a special portion of PACSystems user memory space. On Series 90* and VersaMax* Controllers, you cannot leave the Ref Address property blank.

- **An I/O variable address**, for example, %IX0.6.0.1. (PACSystems only.) This represents the PACSystems Hardware Configuration terminal that contains an I/O variable’s value. For example, this can be a physical discrete or analog I/O point on a module or Genius device, a discrete or analog status returned by the module, or global data.

- **A name**. This makes the current variable an alias variable of the name, that is, a variable that points to the same memory location as the name. The following names may be valid:
  - The name of a parameterized LD block parameter (provided the alias variable is local to the same block)
  - The name and index of an array element, for example, MyComplexArray[255,3]
  - The name of a structure element, for example, MyStructure.MyElement
  - The name of another variable that is not used as an alias variable.

Want to know more? In the online help index, look up “Reference Address (GE IP Controllers)”, “Symbolic Variables”, “I/O Variables”, and “Alias Variables”.

Navigator: Variables tab. This tab displays all the variables used in a project.
Arrays and compound structure variables are supported by Machine Edition*. Variable definitions can be imported from and exported to a variety of file types. You can edit your variables in a spreadsheet and then import them.

Variables used on PACSystems* targets must be externally published in order to be viewed in Proficy® View.

**Want to know more?** In the online help index, look up “Arrays in GE IP Controller targets”, “Structure Data Types”, “Importing variables from a file”, and “Publishing PACSystems* variables”.

### User-defined Data Types

(PACSystems only.) A User-defined Data Type (UDT) is a structured data type consisting of various elements of various other data types. You can define a UDT to reside entirely in discrete memory or non-discrete memory. After defining a UDT, you can create structure variables of that data type and use them in logic.

**Want to know more?** In the online help index, look up “UDT”.

### Variable properties in the Inspector

The Ref Address and other properties of a variable, such as Data Type, are configured in the Inspector. The following is an illustration of the Inspector displaying a typical set of variable properties.
To create a variable

1. In the Variables tab of the Navigator, right-click Variable List, point to New Variable and then choose the data type of the variable. If you have only one target in your project, then Machine Edition creates a new variable with a default name.

If you have more than one target in your project, then the New Variable dialog box appears with a unique default name for the new variable.

2. (Optional.) Enter a unique name for the variable.

   Variable names can range from 1 through 32 characters, begin with a letter or the $ character, contain upper or lower case letters, use numbers between zero and nine, and use the underscore character (“_”).

3. Select the target the variable will be used in.

4. If the new variable is an array, select the Array check box and enter the size(s).

5. Click OK.

   The new variable appears in the list in the Variables tab.

To map a variable to Controller memory or alias a variable

There are two ways to map a variable to Controller memory or alias a variable.

First method:

1. In the Variables tab of the Navigator, right-click a variable and choose Properties.

2. In the inspector that appears, in the Ref Address field, enter a reference address to map the variable or enter the name of another existing variable, a parameterized LD block parameter, an array element, or a structure element to alias the variable.

   - Spell out the exact reference address, for example, %R00123 or 123R (in either case this maps the variable to %R00123), or enter only the memory area, for example %R. This maps the variable to the next available address in that memory area. For example, if %R00122 is the last address used by a 16-bit variable, entering %R maps the variable to %R00123.
- Spell out the name of another variable, or of a parameterized LD block parameter, for example, MyVarWithALongName, or Abc. Or provide the array's name and the element's number, for example, MyArray[5]. Or provide the structure element's name, for example, MyStructure.MyElement.

**Notes (PACSystems* only)**
- If you leave the Ref Address property of a variable blank, the variable is a symbolic variable.
- You can enter an I/O variable address to make the variable an I/O variable, but it is easier to map an I/O variable in the Terminals tab of a module. See “I/O variables”, on page 67.

**Second method to map or alias a variable:**
1. In the Variables tab of the Navigator, right-click a variable and choose **Properties**.
2. In the Inspector that appears, click the **Ref Address** field and then click the **button.

The Reference Address Wizard appears.

3. Do one of the following:
To map a variable to Controller memory: From the **Memory Area** list, choose a region of Controller memory. In the **Index** box, enter an index from the beginning of the region. Select a bit reference. The **Bit Reference** box is available only if you are mapping a BOOL variable to 16-bit memory area on a PACSystems*.

- or -
To alias a variable: In the **Variable Aliasing Filters** section, select the appropriate check boxes for the filters you want to use. In the **Alias variable to** box, select what you want to alias the variable to. Click **OK**.

**Want to know more?** In the online help index, look up “Bit references in non-BOOL variables”.

---

*Proficy* Machine Edition*

**Variables**
**OPTIONS**

The Options tab of the Navigator contains option and preference settings. Options are organized into folders and pages. Click ± or ‏ ‏ to expand or collapse folders. Examples:

- **Controller > General > Duplicate Addresses**: Indicates whether to prevent, warn about, or ignore mapping two variables of the same data size (1, 8, 16, 32, or 64 bits) and length (array size) to the same reference address.

- **Controller > Hardware Configuration > New Reference Assignment**: The way in which default reference addresses are assigned when modules are added to the Hardware Configuration.

- **Editors > Ladder > View > Coil Justification**: The default column in which coils are placed; also the column where the right power rail resides. The default is 10.


**To set options and preferences**

1. In the Options tab of the Navigator, expand an options ‏ ‏ folder and right-click a page within the folder to display the options in the Inspector.

2. In the Inspector, edit the option’s settings as needed.

   **Tip**: When you click an option, the Companion automatically displays help on that option. If the Companion is not already open, press Shift + F11 to open it.

**To reset an entire page of options to their default settings**:

1. In the Options tab of the Navigator, expand an options ‏ ‏ folder.

2. Right-click one of the options pages, and then choose Reset.

   **Tip**: To reset only one option, look up its default value in the Companion and set the option to that value.
MACHINE EDITION HELP

Machine Edition includes a comprehensive online help system that enables you to access specific help topics while working with Machine Edition.

Use the InfoViewer or Companion to access help.

Companion Help

The Companion is a Machine Edition help system that provides useful tips and information while you work. While the Companion is open, it displays help on whatever item is currently selected in the Machine Edition environment.

To use Companion help

1. Ensure that the Companion is open. To open it, press Shift+F11.
2. Click any item in Machine Edition, outside the Companion.

A description of the item you clicked appears in the Companion.

InfoViewer Help

The InfoViewer, another Machine Edition help system, provides detailed information. The InfoViewer has the following features:

- its own toolbar for navigating the help system
- a Table of Contents (in the InfoView tab of the Navigator)
- an index
- a full-text search engine

InfoViewer help is context-sensitive. Click an item on the screen and press F1 to display the appropriate topic in the InfoViewer.
To use the Help index

1. From the Help menu, choose **Index**.

   The Index dialog box appears.

2. Enter a keyword in the top text box or select one in the middle text box.

3. Click **List Topics**.

   A list of topics appears in the bottom text box. The topics are sorted by their rating or likelihood of containing the correct information.

4. In the bottom text box, select a topic to display.

5. Click **Display**.

   The selected Help topic is displayed in the InfoViewer.

   **Tip:** To narrow a search, click the Location heading. The list of topics will be sorted by location and then by rating.
To use the full-text search

You can use the standard HTML Help Viewer to perform a full-text search of the Help. Advanced boolean search expressions can be applied.

1. From the Help menu, choose Search.


2. Enter a search word or phrase in the top text box. SURROUND A PHRASE WITH QUOTATION MARKS.

3. Click to perform a boolean search; that is, use AND, OR, NEAR, or NOT to create a boolean search expression.

4. Click List Topics.

   A list of topics appears in the Select topic box. The topics are sorted by their rating or likelihood of containing the search term, terms, or phrase.

5. In the Select topic box, select a topic to display.

6. Click Display.

   The selected Help topic is displayed in the Help Viewer.

7. (Optional.) Click the InfoViewer button to display the current topic in the InfoViewer.
Tips:

- To narrow the search results, click the Location heading. The listed topics are sorted by location and then by rating.
- When using the HTML Help Viewer window, you may get better results if you select the “Search titles only” check box and/or clear the “Match similar words” check box.

To bookmark favorite topics

1. After using the HTML Viewer to perform a full-text search, select a topic you want to add to your list of favorite topics and then click the Favorites tab.

The Machine Edition Help Search dialog box displays the selected topic title in the Current topic text box (bottom left corner).

2. Click Add to add the topic to the Topics list.

3. (Optional.) Click the InfoViewer button to display the current topic in the InfoViewer.

4. To display a favorite topic in the Help Viewer, select it and click Display.

5. To remove a topic from the Favorites list, select it and click Remove.
To look up topics in the Help Table of Contents

1. Click the InfoView tab of the Navigator. A table of contents for the entire help system appears.
2. Expand Libraries and Books to locate a topic of interest.
3. Double-click a topic.
   The topic is displayed in the InfoViewer.
A target represents a run-time destination of the program or programs you develop with Logic Developer - PLC*. Each target contains all of the components associated with that target. Logic Developer - PLC supports the following GE IP families of Controllers:

- PACSystems* RXi
- PACSystems* RX7i
- PACSystems* RX3i
- Series 90*-70
- Series 90*-30
- VersaMax*
- VersaMax* Nano/Micro
- Series 90* Micro

and the following GE IP remote I/O interface targets:

- Series 90*-30 Ethernet Network Interface Unit
- Series 90*-70 Genius Remote I/O Scanner
- PACSystems* RX3i Ethernet
- VersaMax* Ethernet
- VersaMax* Genius
- VersaMax* Profinet
ADDING, CONFIGURING, AND CONVERTING TARGETS

Adding Targets

Typically a target is present in a project when you use a template to create the project (see page 18). A project can contain multiple targets. Before grouping targets into a project or isolating each target in its own project, consider the following:

- How large are the targets? For example, a target that contains 200,000 variables should probably be by itself in a project.
- Are the targets related? For example, a PACSystems® RX3i with a relatively small program is connected to twenty GE IP Remote I/Os (RIOS). Size is not an issue because RIO targets use little memory in a project. Grouping all the targets into one project is appealing because it makes navigation much easier from one target to another than if the PACSystems RX3i target and RIOs are isolated in 21 separate projects. For another example, if you have various targets that do not communicate with one another or do not serve a common purpose, there may be little value in grouping them in one project.

One target is required for each Controller or remote I/O adapter your project accesses, except when you are using CPU redundancy. In this case, one target contains a Primary Hardware Configuration and a Secondary Hardware Configuration, which correspond to the primary Controller and the redundancy Controller, respectively. For more information, see page 70. Existing targets can be converted from one Controller family to another. For more information, see page 36.

To add a target to a project

1. In the Project tab of the Navigator, right-click the Project node and point to Add Target.
2. Point to GE Intelligent Platforms Controller or GE Intelligent Platforms Remote I/O and then respectively choose the GE IP Controller family or the Remote I/O.

A new target appears in the Project tab of the Navigator. When expanded, the new target contains a default Hardware Configuration, logic blocks, and miscellaneous components.
Configuring Controller Targets

The properties of a target specify the Controller family, the communication connections between your computer and the Controller, and various other settings. All properties are edited in the Inspector. The following table describes common Controller target properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Edit the name for your target in this field.</td>
</tr>
<tr>
<td>Type</td>
<td>The type of target is set by default to GE IP Controller.</td>
</tr>
<tr>
<td>Description</td>
<td>Enter a description of your project in this field. The maximum number of characters is 255.</td>
</tr>
<tr>
<td>Documentation Address</td>
<td>Enter the URL where your project documentation is stored.</td>
</tr>
<tr>
<td>Family</td>
<td>By changing the Controller type in this field, you initiate a target conversion. <strong>Caution: Changes are irreversible.</strong></td>
</tr>
<tr>
<td>Controller Target Name</td>
<td>The name of the target as stored on the Controller.</td>
</tr>
<tr>
<td>Update Rate (ms)</td>
<td>Set the rate at which the screen is updated while online to the target.</td>
</tr>
<tr>
<td>Sweep Time (ms)</td>
<td>The sweep time of the Controller when online. This value is also displayed on the status bar. (Read-only.)</td>
</tr>
<tr>
<td>Controller Status</td>
<td>The online/offline, run/stop status of the Controller. (Read-only.)</td>
</tr>
<tr>
<td>Online Project Synchronization</td>
<td>When Enabled, Machine Edition performs project synchronization checks with an online Controller.</td>
</tr>
<tr>
<td>Physical Port</td>
<td>Choose the type of connection to the Controller (Ethernet, COM, or modem).</td>
</tr>
<tr>
<td>IP Address</td>
<td>(Ethernet protocol only.) Set the IP address of the Controller.</td>
</tr>
<tr>
<td>Additional Configuration</td>
<td>Group of properties used for the detailed configuration of your communication connection.</td>
</tr>
<tr>
<td>Modem Communication</td>
<td>Group of properties to configure the modem communications link between your computer and a remote GE target.</td>
</tr>
</tbody>
</table>

Other properties are available depending on the value of the Family property and the CPU model.

To configure a Controller target

1. In the Project tab of the Navigator, right-click a target and choose **Properties**.
   The properties of the target appear in the Inspector.
2. In the Inspector, configure the properties as needed.
Converting Targets

With Logic Developer - PLC*, you can convert targets from one GE IP Controller family type to another. For example, you can take the Hardware Configuration and logic written for a Series 90*-70 Controller and convert them for use on a PACSystems* RX7i. However, target conversions are irreversible; when logic blocks are deleted during a conversion, they cannot be restored. We recommend that you make a backup of your project before converting a target. There may also be logic execution differences when converting an application from one GE IP Controller family type to another, especially if a Series 90-70 is involved in the conversion. You must validate the application execution prior to deployment into a production environment.

To convert a target

1. In the Project tab of the Navigator, right-click a target and choose Properties.
   The Inspector displays the target's properties.
2. In the Inspector, click Family.
3. From the list, choose the new GE IP Controller family you want to convert the target to.

![Inspector](image)

<table>
<thead>
<tr>
<th>Target</th>
<th>Name</th>
<th>S9070</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>GE Controller</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Documentation Address</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Family</td>
<td>Series 90-70 PLC</td>
<td></td>
</tr>
<tr>
<td>Controller Target Name</td>
<td>PACSystems RX7i</td>
<td></td>
</tr>
<tr>
<td>Update Rate (ms)</td>
<td>PACSystems RX7i</td>
<td></td>
</tr>
<tr>
<td>Swap Time (ms)</td>
<td>Series 90-70 PLC</td>
<td></td>
</tr>
<tr>
<td>Controller Status</td>
<td>Series 90-70 PLC</td>
<td></td>
</tr>
<tr>
<td>Enable Shared Variables</td>
<td>Series 90-70 PLC</td>
<td></td>
</tr>
<tr>
<td>Physical Port</td>
<td>COM1</td>
<td></td>
</tr>
<tr>
<td>Additional Configuration</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
There are two types of target conversions.

A **basic** conversion does the following:

- Strips all the configured modules from the original Hardware Configuration (HWC) and sets up the destination family's default HWC, with a single power supply and a CPU.
- Deletes or adds target components, logic programs, or blocks of logic.
- Updates the system variables, including the fault locating references if applicable.

An **enhanced** conversion does the following:

- Replaces a power supply with the destination family's default power supply.
- Retains the settings of CPU parameters common to the original and destination CPUs when the settings are supported in the destination family; otherwise, replaces them with the default settings.
- Retains all the original modules that are supported by the destination family. Retains all the module parameter settings that are supported by the destination family; otherwise, the default settings are used.
- Ethernet module settings of non-supported Ethernet modules are used to configure the destination's Ethernet daughterboard or default Ethernet modules.
- Converts unsupported expansion racks to the nearest equivalents.
- Deletes or adds target components, logic programs, or blocks of logic.
- Updates the system variables, including the fault locating references if applicable.
- Displays a conversion report in the InfoViewer.

**Caution:** The capabilities of a destination target may be different from those of the original target. Carefully examine the conversion report when it is available. Validate the project and test it thoroughly before deploying it in production.

**Note:** The hardware must be configured before it is operational. For detailed information on hardware configuration, see page 62.

*Want to know more?* In the online help index, look up "Target Conversions".
CONFIGURING COMMUNICATION

For Logic Developer - PLC* to communicate with a target Controller, a connection must be properly configured. The properties of a target are adjustable to accommodate your connection(s).

To configure an Ethernet, modem, or serial connection with any GE IP Controller

1. In the Project tab of the Navigator, right-click a target and choose Properties.
   The Inspector displays the target’s properties.

2. In the Inspector, set the Physical Port property to Ethernet or a COM port (or a modem if one is installed).

3. If the Physical Port is Ethernet, enter the IP Address of the target Controller.

4. Double-click Additional Configuration to access the detailed settings for your connection.

Note: An IP address must be set in the Controller before an Ethernet connection can be established.

*Want to know more?* In the online help index, look up "Configuring communications with GE IP Controller targets".
To set a temporary IP address for a PACSystems® RX7i, PACSystems® RX3i, or VersaMax®
Ethernet NIU target

1. In the Utilities tab of the Navigator, double-click Set Temporary IP Address.
The Set Temporary IP Address dialog box appears.

2. In the MAC Address field, specify the MAC address for the target.

3. In the IP Address to Set field, specify the temporary IP address you want to set on the target.

4. If required, select the Network Interface Selection check box and specify the network interface the target is located on.

5. Click the Set IP button.
The IP address of the specified Controller or device is temporarily set to the indicated address.

Warning: The IP address set by the Set Temporary IP Address utility is not retained through a power cycle. If you want to set a permanent IP address, you must download the Hardware Configuration to the Controller.
To set a permanent IP address for a PACSystems*

1. In the ⌍ Project tab of the ⌍ Navigator, right-click the 🏛 target and choose Properties.
   The ⌍ Inspector displays the target’s properties.
2. In the IP Address property, enter an IP address.
3. Expand the 🏛 Hardware Configuration.
4. Do one of the following:
   - For a PACSystems* RXi, expand the PACSystems RXi node, and double-click the Ethernet node.
   - For a PACSystems* RX7i, expand the ⌍ main rack, expand the ⚪ CPU, and double-click the Ethernet subnode.
   - For a PACSystems* RX3i, expand the ⌍ main rack, double-click every ⚪ IC695ETM001 Ethernet module and repeat step 5 for each one of them.
5. In the Settings tab that appears in the Parameter editor, set the IP Address, the Subnet Mask, and the Gateway IP Address.
6. Download the Hardware Configuration to the PACSystems.

Want to know more? In the online help index, look up “Configuring communications with GE IP Controller targets” and “Set Temporary IP Address utility”.

To set an IP address for Series 90*-70 Controllers

1. In the ⌍ Project tab of the ⌍ Navigator, expand the 🏛 Hardware Configuration.
2. Expand the 🏛 main rack and double-click the 🛋 slot where you want to add a communications module.
The Module Catalog appears.

<table>
<thead>
<tr>
<th>Module Catalog</th>
</tr>
</thead>
<tbody>
<tr>
<td>Catalog Number</td>
</tr>
<tr>
<td>IC697CMM741</td>
</tr>
<tr>
<td>IC697CMM742</td>
</tr>
<tr>
<td>IC697BE4703</td>
</tr>
</tbody>
</table>

3. Choose the Communications tab and select the communications module you want to add.
   Select either IC697CMM741 or IC697CMM742.
4. Click **OK**.
   The Parameter editor displays the configuration settings for the module.
5. In the Settings tab, double-click the **IP Address** parameter.
   The IP Address dialog box appears.
6. Enter the IP Address and click **OK**.
7. Download the Hardware Configuration. For more information, see page 47.

**To set an IP address for a Series 90*-30 CPU364, CPU372, CPU374, NIU004, or a VersaMax* CPUE05**

**Note:** An Ethernet adapter is built into CPU364, CPU372, CPU374, NIU004, and CPUE05. For other Series 90*-30 CPUs, Ethernet connections are supported through an optional expansion slot on the rack.

1. Obtain an IP address, perhaps from your network administrator.
2. In the **Project** tab of the **Navigator**, expand the **Hardware Configuration** and expand the main rack.
3. Right-click the slot containing the CPU364, CPU372, CPU374, NIU004, or CPUE05 and choose **Configure**.
4. In the Parameter editor that appears, click the Ethernet tab.

The Parameter editor displays the contents of the Ethernet tab.

![Parameter Editor](image)

5. Double-click the IP address parameter.

The IP Address dialog box appears.

6. Enter the IP address in the box and click OK.

7. Download the Hardware Configuration. For more information, see page 47.

To set an IP address for other Series 90*-30 CPUs (not CPU364, CPU372, or CPU374)

1. In the Project tab of the Navigator, expand the Hardware Configuration.

2. Right-click the slot containing an Ethernet Interface Module (IC693CMM321) and choose Configure.

   The Parameter editor displays the configuration settings for the module.

3. In the Settings tab, double-click the IP Address field.

   The IP Address dialog box appears.

4. Enter the IP address and click OK.

5. Download the hardware configuration. For more information, see page 47.

To download an IP address via a Controller serial port

1. Configure a CPU or Ethernet communications module with an IP address using Hardware Configuration.

2. Right-click the target and choose Properties.

   The Inspector displays the target’s properties.

3. In the Physical Port property, select a serial connection.

4. Right-click the target and choose Go Online.
5. Right-click the target and choose **Download to Controller**.
   The Download to Controller dialog box appears.

6. Select **Hardware Configuration** and click **OK**.
   The IP address is assigned to the Controller and the rest of the Hardware Configuration is downloaded to the Controller.

7. Right-click the target and choose **Go Offline**.

8. Right-click the target and choose **Properties**.

9. In the Physical Port property, select **Ethernet**.
   The next time you go online, Machine Edition* will use an Ethernet connection with the specified IP address.
INTERACTING WITH A CONTROLLER

Communicating with a PACSystems®, Series 90, or VersaMax® is necessary to perform such operations as storing and loading programs or monitoring data values and Controller Status information. You can connect to a Controller from Logic Developer - PLC® over a serial, Ethernet, or modem connection, depending on the Controller’s capabilities.

All interactions with a target are available from the target’s right-click menu.

Validating a Target

Validating your target detects syntax and configuration errors on the target. Error messages are generated for each error and displayed in the Feedback Zone. A target containing errors cannot be downloaded. A target is automatically validated before a download operation.

To validate a target

- In the Project tab of the Navigator, right-click a target and choose Validate.

Logic Developer - PLC® checks all items under the target for syntax errors. Any errors detected are noted in the Build tab of the Feedback Zone.

Tip: Double-click an error message to locate the noted error in your project. The appropriate editor or tool opens automatically with the item in question selected. Press the F4 key to locate the next error or warning in your project. Tips showing you how to proceed are displayed in the Companion. To open the Companion, press SHIFT + F11.

Offline, Online: monitor mode, programmer mode

When offline from a PACSystems®, Series 90, or VersaMax®, there is no ongoing communication between the Controller and your development computer. A physical communication link is not required as long as you only edit logic; it is required only when you want to communicate with the Controller.

The only Controller operations you can perform while offline are to go online or to set up the temporary IP address (see page 39).

When online with a target Controller, a communication link exists and is active and you have an ongoing real-time communication with the Controller.
When online in monitor mode, you can monitor the Controller while it is executing. You cannot edit logic on your computer except State Logic. You cannot change any values on the Controller. Depending on your level access on the Controller and your Change Management permission levels, you can upload from the Controller.

When online in programmer mode, you can make changes on your computer and the Controller and can monitor the Controller while it is executing. You can edit any type of Controller logic on your computer. Depending on your level access on the Controller and your Change Management permission levels, you can upload from the Controller, control the Controller while it is executing, and change values on the Controller, including downloading to the Controller. When editing LD logic while online, you can make word-for-word changes (see page 102); on PACSystems, you can also use the test edit feature (see page 100).

When online in either mode, if the project stored on the Controller is equal to the current project in Logic Developer - PLC*, the LD editor displays a graphical representation of LD logic as it executes.

To go online to a GE IP Controller

1. In the Project tab of the Navigator, ensure that no target Controller is already online.
   You can be online to only one target at a time. When a target is offline, its icon is grey.

2. If the name of the target you want to go online to does not appear in bold characters, right-click the target and choose Set as Active Target. Bold characters indicate that this is the active target.

3. Do one of the following:
   - Right-click the active target and choose Go Online.
   - or -
   - Click the thunderbolt on the Online toolbar:

Logic Developer - PLC* connects your project to the GE IP Controller. The online status is indicated by the target icon in the Project tab and on the status bar.
Note: When online to a GE IP Controller, the target icon in the Project tab of the Navigator appears as  (equal),  (not equal), or  (stop faulted).

To change the online mode

- In the Project tab of the Navigator, right-click the target Controller, point to Online Commands, and choose Set Programmer Mode or Set Monitor Mode.

To go offline from a GE IP Controller

- Do one of the following:
  - In the Project tab of the Navigator, right-click a target Controller and choose Go Offline.
  - or -
  - Click the thunderbolt on the Online toolbar: Logic Developer - PLC goes offline.

Want to know more? In the online help index, look up “going online to GE IP Controllers”, “setting online programmer mode”, “setting online monitor mode”, and “going offline from GE IP Controllers”. 
**Upload/Download**

The download process builds and validates all run-time files necessary for a target to perform its role in a completed project. The compiled project is then transferred to the target hardware over the communication connection previously configured. For more information, see page 38.

The upload process acquires a project from the active GE IP Controller target and transfers it to Logic Developer - PLC for editing.

**To download to a GE IP Controller**

1. Ensure that you are online in programmer mode to the target Controller.
2. In the Project tab of the Navigator, right-click the target to which you want to download files and choose **Download to Controller**. The Download to Controller dialog box appears, displaying these or other options.

   ![Download to Controller dialog box](image)

   **Note:** If the Controller is running, you can download only logic that is not equal to the Controller’s current logic and the Download to Controller dialog box does not appear. On a PACSystems®, some source files that do not affect equality are also downloaded.

3. Choose the items you want to download and click **OK**.

   **Note:** Only one project can be downloaded to a target at a time. If you download to a target Controller that already has a project on it, the existing project is overwritten.

   For each target that you download, Machine Edition* performs a validation. Any errors that occur are displayed in the Build tab of the Feedback Zone. If there are no errors, Machine Edition builds and sends all the necessary run-time files to the Controller.

   *Want to know more?* In the online help index, look up “downloading to GE IP Controllers”.  

---

**GE IP Targets**

*Interacting with a Controller*
To upload files from a GE IP Controller

1. Ensure that you are online to the target Controller.

2. In the Project tab of the Navigator, right-click the target Controller from which you want to upload information and choose Upload from Controller.

   The Upload from Controller dialog box appears, displaying these or other options.

3. Choose the item(s) you want to upload and click OK.

   The selected items are uploaded to Logic Developer - PLC*. If you already had a version of the project open, the uploaded project merges with the existing project. Because variable names are not stored on Series 90* or VersaMax* Controllers, if you upload to an empty target, all variables are assigned default names. For example, %R00001 is named R00001.

   Want to know more? In the online help index, look up "uploading from GE IP Controllers".
Run/Stop

When you are online in programmer mode, you can set a target GE IP Controller to Run or Stop mode. In Stop mode, you can choose to enable or disable the outputs.

To start a GE IP Controller

- Do one of the following:
  - In the Project tab of the Navigator, right-click a target, point to Online Commands, and then choose Start Controller.
  - or -
  - Click on the Online toolbar:

    ![Start Controller button]

    The target Controller begins executing its program.

Note: If you are starting a PACSystems* or Series 90*-70 Controller, you can choose to have outputs enabled or disabled.

To stop a GE IP Controller

1. Do one of the following:
   - In the Project tab of the Navigator, right-click a target, point to Online Commands, and choose Stop Controller.
   - or -
   - Click on the Online toolbar:

    ![Stop Controller button]
The Stop Controller dialog box appears, prompting you to enable or disable the Controller’s outputs.

2. Select an option.
3. Click OK.

The target Controller stops executing its program.

Fault Tables

The Controller and I/O Fault Tables display fault information logged by the CPU or modules in the Controller. This information is used to determine if there are problems with the Controller hardware or software running in the Controller’s CPU.

To view the fault table reports

- Do one of the following:
  - In the Project tab of the Navigator, double-click the target you want a fault table report on, or right-click it and choose Diagnostics.
  - Or -
  - In the status bar, at the bottom of the Machine Edition* window, double-click the target.

Note: To view the Controller and I/O fault tables, your computer must be online to the Controller. To clear faults, you must be in online programmer mode.
The fault table appears in the InfoViewer window.

- Indicate online status of Controller.
- View the date and time of the faults.
- Double-click a row to view details of faults.

Want to know more? In the online help index, look up “fault” (in lower case), then select “Working with the Fault Tables”.

Choose the type of fault information you require.
Print fault table by clicking here.
Save fault table to xml file by clicking.
Select the format of fault data.
Sort fault information.
(When in online programmer mode only.) Clear fault information by clicking here.

Fault Table Viewer

Controller Fault Table (Displaying 16 of 81 faults, 55 Overflown)

Loc Code/状

Fault Description
Date/Time
1
Can’t locate remote node: discarded request
09-23-2000 01:50:47
2
Can’t locate remote node: discarded request
09-23-2000 02:40:09
3
Bad remote application request: discarded request
09-23-2000 02:30:56
4
Connection to remote node failed: assuming act
08-17-2000 03:38:53
5
Can’t locate remote node: discarded request
08-26-2000 23:51:49

Error Code

Group

Action

Task/Sourc

Fault Description

Date/Time

1
LAN severe network problem: attempting recovery
08-26-2000 23:50:49
2
Local request to send was rejected: discarded request
08-26-2000 23:18:21
3
LAN severe network problem: attempting recovery
08-26-2000 22:42:34
4
Reset of communications interface
01-01-2000 03:02:12
5
No user program present
01-01-2000 03:02:09

GFK-1918R

Proficy* Logic Developer - PLC* 8.00
Reference View Tables

In Reference View Tables (RVTs), you can monitor reference data when you are online to a GE IP Controller. If you are in online programmer mode, you can also use RVTs to change the values of reference data. In the Project tab of the Navigator, the Reference View Tables folder contains a Default Tables folder. You can add user-defined tables to the Reference View Tables folder. A target can have zero or more user-defined RVTs.

You can configure the default appearance of your RVTs in the Options tab of the Navigator. For more information, see “Options”, on page 26.

Data values at sequential addresses are displayed from right to left, by default, starting at the reference address specified in the Address column. Both default and user-defined RVTs display rows of 8 cells for discrete memory (each cell corresponding to 8 bits) and rows of 10 cells for register memory (each cell corresponding to one 16-bit register). The amount of data displayed in the columns depends on the data display format.

To create a user-defined Reference View Table

- In the Project tab of the Navigator, right-click the Reference View Tables folder and choose New.

A new Reference View Table with a default name is added to the folder.
To work with a user-defined Reference View Table

1. In the Project tab of the Navigator, expand the Reference View Tables folder and double-click the table you want to view.

   The Reference View Table appears in the main Machine Edition window.

2. Add reference addresses to the table as required.

   Note: You cannot add reference addresses to a default RVT.

3. Format the table entries as desired.

---

Want to know more? In the online help index, look up “RVT”, then select “Working with user-defined Reference View Tables” or “Working with default Reference View Tables”.

---

Reports

Reports provide summaries and tables of information about your project. Most reports are displayed in the InfoViewer (see page 27). The Reports tab of the Feedback Zone contains a list of all reports generated since the last Machine Edition project was opened. The following are types of reports and logic printouts available in Logic Developer - PLC:

- Address Use report
- Application Structure reports
- CAM profile report
- GGD reports
- Forces in Controller report
- Hardware Configuration report
- Initial Force States in Project report
- Modbus Address report
- Variable Difference report
- Variables reports
- Structured Text Blocks

An L indicates a logic printout.

To generate reports

- In the Project tab of the Navigator, right-click a node and choose Report to generate a report on that node.

A report is automatically generated and displayed in the InfoViewer.

Note: To generate a Forces in Controller report for a target, you must be online to the target. Right-click the target and choose Report. In the Select a Report dialog box, select Forces in Controller Report and click OK.
To redisplay a previously generated report

1. In the Feedback Zone, double-click the Reports tab.
   
   A list of previously generated reports appears in the Feedback Zone.

2. In the list, click the report you want to view.
   
   The report appears in the InfoViewer.

   Many items in a report contain hyperlinks. Click a hyperlinked item to jump to that item in the project. For example, if a variable's name appears hyperlinked in a report, clicking it selects that variable in the Variables tab of the Navigator.

   Large reports are often separated into several pages. To view a different page of the report, scroll to the bottom of the report in the InfoViewer and click the number of the page you want to view.

To print a report displayed in the InfoViewer

1. Generate the report you want to print, or redisplay a previously generated report.

2. When the report is displayed, right-click the InfoViewer window and choose Print.

To print LD Blocks

1. In the Project tab of the Navigator, expand the Logic node.

2. Right-click the Program Blocks node and choose Print LD Blocks.
   
   The Print dialog box appears.

3. Select the blocks to print.
   
   - To print all of the target's LD blocks in alphabetical order, select the All option.
   
   - To print only some of the target's LD blocks, select the Selection option, then select the check box in front of each block you want to print. To determine in which order to print the blocks, select them one at a time and click Up or Down until the selected blocks appear in the desired order.

4. Select options as required and click OK.

To print ST blocks

1. In the Project tab of the Navigator, expand the Logic node.

2. Right-click the Program Blocks node and choose Report of ST Blocks.
   
   The Structured Text Blocks logic printout is automatically generated and displayed in the InfoViewer.

  "Want to know more?" In the online help index, look up "Reports" and "printing ST logic".
Hardware Configuration

Logic Developer - PLC* supports several GE IP Controller families and various GE remote I/O interfaces (see page 80) with a variety of CPUs, racks, and modules for each. In order to operate, GE IP Controller hardware must be configured with Logic Developer - PLC or some other GE tool. The HWC component of Logic Developer - PLC provides a way to configure your target equipment. This chapter details specifics on configuring Controller hardware for your operational needs.

The first step in configuring Controller hardware is to select the Controller you want to configure (see page 34). When creating a new project, you can use a project template containing a default Hardware Configuration, or you can create an empty project and configure it manually.

PACSYSTEMS* RXI

The PACSystems* RXi, pictured below, is an advanced, high performance, small footprint, PROFINET-dedicated Controller designed for distributed applications (process or discrete) in rugged environments.
The PACSystems® RXi has the following characteristics:

- 2-port (shared MAC) Gigabit Ethernet I/O PROFINET network connection
- Additional 1xGB Ethernet port
- Built-in cable redundancy (MRP) delivering PROFINET IO cabling redundancy with no external switches
- Dual core COMExpress ICRXICL000 CPU with 2 GB RAM and up to 64 GB internal flash data storage
- Support for most LD, FBD, and ST instructions supported on other PACSystems platforms
- High speed Interconnect Bus enabling combinations of control and Proficy (or other Microsoft® Windows® or Linux applications)
- Internal industrial grade SSD drive providing local long-term data retention
- Energy Pak supplying power to write data to NV RAM during power failures
- USB and SD interfaces enabling program loading, serial communications, and data storage via standard devices
- (Optional.) Intelligent Display Module, a configuration and maintenance touchscreen right on the Controller

Want to know more? From the Help menu, choose Important Product Information. In the window that appears, select the Logic Developer - PLC product, select Version 7.50, select the New Features tab, and click PACSystems RXi.
The PACSystems RX7i is pictured below.

The PACSystems RX7i consists of a main rack and up to seven expansion racks. The PACSystems RX7i supports a few types of main racks. It supports five types of Series 90*-70 expansion racks. All main racks and expansion racks support a few non-intelligent Series 90-70 fan kits.

The PACSystems RX7i main racks have 10 or 18 single-width VME slots. Slot 0 supports one of the power supply modules. Slots 1 and 2 are used to house a double-width CPU. PACSystems RX7i supports a few CPU models. Each PACSystems RX7i CPU has an embedded Ethernet adapter. In addition to the Ethernet adapter, up to seven IC698ETM001 Ethernet modules can be installed in a PACSystems RX7i main rack. An Ethernet module occupies only one slot. All of the IC698... modules are supported only on the main rack.

PACSystems RX7i supports most Series 90*-70 modules. Most Series 90-70 modules are double-width, that is, they occupy two slots on the main rack, but they occupy only one slot on a Series 90-70 expansion rack. You can plug double-width modules into any two adjacent slots of the main rack except slots 0, 1, or 2. Because slot 17 is a double-width slot, any module in slot 17 occupies only slot 17. Slot 17 can also accept a single-width module.

PACSystems RX7i supports the following Series 90-70 bus expansion modules:

- IC697BEM711
- IC687BEM713
- IC697BEM713
PACSystems RX7i also supports Series 90-70 Genius Bus Controllers, which support numerous Genius devices.

**Configuring PACSystems RX7i Hardware**

When you create a target with a PACSystems RX7i rack system, the default Hardware Configuration consists of the main rack, with a power supply in slot 0 and a single CPU in slots 1 and 2. You can replace the default power supply and CPU, and add single-width or double-width modules. You can add up to seven expansion racks, and on each of these, you can add Series 90-70 modules.

**Note:** On PACSystems RX7i targets, only one rack is added by default. On Series 90-70 and Series 90*-30 Controllers, seven expansion racks are added by default to the HWC. You do not need to add them.
PAC SYSTEMS* RX3I

The PACSystems RX3i, pictured below, is an upward compatible upgrade for Series 90*-30 Controllers, offering more memory, faster clock speed, and an enhanced backplane bus.

The PACSystems RX3i consists of a main rack and up to seven expansion racks. Six types of Series 90-30 expansion racks are supported; they have IC693CHSnnn catalog numbers and are colored black, in keeping with the Series 90-30 color scheme. Two expansion racks have IC694CHSnnn catalog numbers. These are identical to the IC693CHSnnn expansion racks of the same numbers except that they are colored blue, in keeping with the PACSystems RX3i color scheme.

The main rack and all expansion racks support most Series 90-30 modules; these have IC693NNNnnn catalog numbers and are colored black, in keeping with the Series 90-30 color scheme. Almost all of the IC693NNNnnn modules supported by PACSystems RX3i also come as IC694NNNnnn modules that have identical functionality. The only difference is that the IC694NNNnnn modules are colored blue, in keeping with the PACSystems RX3i color scheme. For example, the IC693DSM314 and IC694DSM314 are functionally identical and both can be used interchangeably in a Series 90-30 rack system or a PACSystems RX3i rack system. The only hardware difference is their color.

When you configure a PACSystems RX3i rack system, you can select IC693NNNnnn or IC694NNNnnn racks or modules. When you configure a Series 90-30 rack system, however, you can select only IC693NNNnnn racks or modules. Even if you are physically using IC694NNNnnn racks or modules, selecting IC693NNNnnn racks or modules in Logic Developer - PLC* is adequate.
Hardware Configuration

PACSystems* RX3i

The PACSystems* RX3i default main rack, IC695CHS012, has 13 slots: Slots 0 through 12. The alternate main rack, IC695CHS016, contains 17 slots: Slots 0 through 16.

On either main rack, slot 0 is reserved for a power supply or the CPU, but the power supply or CPU does not have to be in slot 0. The last slot is reserved for the Serial Bus Transmitter module (IC695LRE001). The slot just before the last slot can contain any single-width module native to PACSystems RX3i (IC695...) except the Serial Bus Transmitter. Any other slot can contain any single-width module native to PACSystems RX3i (IC695...) except the Serial Bus Transmitter, and if the next slot is empty, they can contain any supported double-width module.

The IC695PSA040 and IC695PSA140 power supplies and the CPU are two-slot modules. The IC695PSD040 and IC695PSD140 power supplies are one-slot modules. PACSystems RX3i supports many Series 90*-30 modules: each one occupies a single slot.

Configuring PACSystems* RX3i Hardware

When you create a target with a PACSystems RX3i rack system, the default HWC consists of the main rack, with a power supply in slots 0 and 1, and a CPU in slots 2 and 3. You can replace the default power supply. You can replace the CPU with itself to update it to the latest catalog version, that is, the latest set of parameters supported for the CPU in Logic Developer - PLC*. You can move the power supply and the CPU to any empty slot in the target with an adjacent empty slot; however, the second adjacent slot cannot be the last slot. You can add up to seven expansion racks, and on each of these, you can add IC693NNNnnn modules and the IC694NNNnnn that are supported by Series 90*-30.

Note: On PACSystems RX3i targets, only the main rack is added by default. On Series 90*-70 and Series 90*-30 Controllers, seven expansion racks are added by default to the HWC. You do not need to add them.
CONFIGURING CONTROLLER HARDWARE

The following configuration procedures focus on the PACSystems® RX3i. Procedures for the other Controller families supported by Logic Developer - PLC® are nearly identical, when applicable.

The default power supply for the PACSystems RX3i is the IC695PSA040.

To replace a power supply (not applicable for PACSystems RXi)

1. In the Project tab of the Navigator, right-click the power supply slot and choose Replace Module.

A list of available power supplies appears.

Note: For non-PACSystems GE IP Controller families, there is no Slot 0. Right-click the PWR slot instead.

2. Select the power supply you have installed in your rack and click OK.

The default CPU specified in the project template for a PACSystems RX3i is the IC695CPU320.

To replace a CPU (not applicable for PACSystems RXi)

1. In the Project tab of the Navigator, expand the Hardware Configuration.

All racks are revealed.

2. Expand the main rack.

3. Right-click the CPU slot and choose Replace Module.

Note: A GE IP Controller supports only one CPU; it can be replaced with itself to update it to the latest catalog version, that is, the latest set of parameters supported for the CPU in Logic Developer - PLC®.
4. From the list, select a CPU.
5. Click OK.
A dialog box appears asking if you want to retain the settings from the existing CPU.
6. Click Yes or No.
The target is configured with the selected CPU.

**To configure a CPU**

1. In the Project tab of the Navigator, right-click a slot containing a CPU and choose Configure.
The Parameter editor displays all configurable settings for the CPU.
2. Modify the settings as required. For information on any parameter of a PACSystems* CPU, select the parameter. Help topics for PACSystems CPU parameters appear in the Companion. To open the Companion, press SHIFT+F11.

**To add an expansion rack (PACSystems* RX7i and PACSystems* RX3i only)**

**Note:** On Series 90*-70 and Series 90*-30 Controllers, seven expansion racks are added by default to the HWC. You do not need to add them.

1. In the Project tab of the Navigator, right-click the Hardware Configuration and choose Add Rack.

   The Catalog dialog box appears listing available rack types.

2. Select a rack and click OK.

   The Series 90*-30 expansion rack is added to the PACSystems* RX3i rack system. It is assigned an available number in the range of 1 through 7, excluding numbers already assigned to other expansion racks.

**To replace a rack (not applicable for PACSystems RXi)**

1. In the Project tab of the Navigator, right-click a rack and choose Replace Rack.

   The Catalog dialog box appears listing available rack types.

2. Select a rack and click OK.
To add a fan kit to a rack (PACSystems® RX7i only)

1. In the Project tab of the Navigator, expand the Hardware Configuration.

2. Right-click a rack and choose Properties. The Inspector displays the rack's properties.

3. In the Inspector, click the Fan Catalog Number property. The button appears to its right.

4. Click the button. The Catalog dialog box appears.

5. Select the fan kit you want to add and click OK. The fan kit is added to the rack.

To add an Ethernet module (PACSystems® RX7i and PACSystems® RX3i only)

1. In the Project tab of the Navigator, expand the Hardware Configuration and then Rack 0 (the main rack).

2. Double-click the slot you want to add an Ethernet module to, or right-click it and choose Add Module.
The Catalog appears.

3. In the Communications tab of the Catalog, select the Ethernet Module and click **OK**.

   The Ethernet module is added to the slot.

4. Double-click the Ethernet module.

   The Parameter editor appears.

5. Configure the Ethernet module's parameters as needed.

   "Want to know more?" In the online help index, look up "configuring PACSystems RX3i rack system" and "Parameter editor features".
Hardware Configuration
Configuring Controller Hardware

To configure the Ethernet daughterboard (PACSystems* RX7i and RX3i only)
1. In the Project tab of the Navigator, expand the Hardware Configuration, then Rack 0 (the main rack), and then the CPU.
2. Double-click Ethernet.
   The Parameter editor appears.
3. Configure the Ethernet daughterboard’s parameters as needed.

To move a module (not applicable for PACSystems* RXi)
1. In the Project tab of the Navigator, expand the Hardware Configuration and then the rack that contains the module.
2. Do one of the following:
   - To move the module from one target to another: press SHIFT while dragging and dropping the module onto an appropriate empty slot.
   - Or -
   - To move the module within a target: drag and drop the module onto an appropriate empty slot without pressing any keys.

You cannot move modules between different types of Controller targets, for example from a Series 90*-30 to a PACSystems* RX7i.

The module is removed from the original slot and inserted into the empty slot.

Notes: (PACSystems RX7i and PACSystems RX3i)
- You can drag a double-width module to a main rack slot only if both the slot and the slot after it are empty. Exceptions: The last slot on a PACSystems* RX7i can contain a double-width module or one single-width module. The last slot on a PACSystems RX3i main rack can contain only an IC695LRE001.
- Only PACSystems RX3i supports moving a power supply or CPU module.
To copy a module

1. In the Project tab of the Navigator, expand the Hardware Configuration and then the rack that contains the module.

2. Do one of the following:
   - To copy the module from one target to another: drag and drop the module onto an appropriate empty slot without pressing any keys.
   - or -
   - To copy the module within a target (not applicable for PACSystems RXi): press CTRL while dragging and dropping onto an appropriate empty slot.

You cannot copy modules between different types of Controller targets, for example from a Series 90*-30 to a PACSystems RX7i. Exception: You can copy a PROFINET Controller from a PACSystems RX3i to a PACSystems RXi or vice-versa.

A copy of the module is inserted into the empty slot.

Notes

- Only the PROFINET Controller can be copied to or from a PACSystems RXi.
- (PACSystems RX7i and PACSystems RX3i.) You can drag and drop a double-width module to a main rack slot only if both the slot and the slot after it are empty. Exceptions: The last slot on a PACSystems RX7i can contain a double-width module or one single-width module. The last slot on a PACSystems RX3i can contain only an IC695LRE001.
- Only PACSystems RX3i supports copying a power supply.

I/O VARIABLES

An I/O variable is a variable mapped to a terminal in the Hardware Configuration of a PACSystems*. A terminal, for example, can be one of the following:

- A physical I/O discrete or analog point on a PACSystems module or on a Genius device
- A discrete or analog status returned from a PACSystems module
- Genius global data

Memory required to support I/O variables counts against your user memory. When you configure the PACSystems CPU (see page 62), select the Memory...
Hardware Configuration

I/O variables

tab and set the I/O Discrete (# of Bits) and I/O Non-Discrete (# of Words) parameters to configure the space available for I/O variables.

To enable I/O variables for a module

1. In the Project tab of the Navigator, expand the target that contains the module.
2. Expand the Hardware Configuration and expand the rack that contains the module.
3. Right-click the module and choose Properties. The Inspector displays the module’s properties.
4. In the Inspector, set the Variable Mode property to True.
   If the Variable Mode property is read-only and set to False, the module does not support I/O variables.
5. In the dialog box that appears, click Yes.
   If you enable I/O variables for a Genius bus Controller, I/O variables are enabled for all the Genius devices on the Genius bus.
   Note: When you enable I/O variables for a module, the Hardware Configuration and logic become coupled. This means that they must be downloaded, uploaded, or cleared together. You cannot change or add I/O variables in Run Mode Store.

To map a variable to a terminal in the Terminals tab of a module or Genius device

1. In the Project tab of the Navigator, expand the target that contains the module or Genius device.
2. Expand the Hardware Configuration and expand the rack that contains the module or device.
3. Double-click the module or expand the Genius Bus Controller (GBC) and double-click the Genius device.
The Parameter editor displays the Terminals tab of the module or device.

4. In the Terminals tab, right-click a terminal node that has no I/O variable mapped to it and choose Map Variable. The Variables smart list appears.

5. In the smart list, do one of the following:

- Enter a name that is not used elsewhere in the PACSystems* target. A new I/O variable by that name is created and mapped to the terminal. Its Ref Address property is set to an I/O variable address. If the terminal is discrete, the new I/O variable’s data type is set to BOOL. If the terminal is analog, the data type is set to INT, but you can change it to another 16-bit data type: UINT or WORD.

- or -

- Enter the name of an existing CPU-mapped variable, symbolic variable, or alias variable. (There are some mapping limitations.) The variable becomes an I/O variable mapped to the terminal and its Ref Address property is changed accordingly.

Want to know more? In the online help index, look up “I/O Variables (PACSystems)*”, “working with I/O variables (PACSystems)*”, and “Coupled Hardware Configuration and Logic”.
**HOT REDUNDANCY SYSTEMS**

Hot redundancy systems are supported only on PACSystems* RX7i, PACSystems* RX3i, and Series 90*-70 Controllers. In redundancy systems, two units are set up and configured identically. If one unit fails or is taken offline, the other unit assumes responsibility without interrupting operation of the entire system.

There are three types of hot redundancy systems:

- Basic CPU Redundancy
- Genius Redundancy
- CPU Redundancy Over Genius

**Basic Hot CPU Redundancy**

To implement basic hot CPU redundancy, a single Logic Developer - PLC* target is associated with two physical Controllers, a Primary and a Secondary. Both Controllers share the same logic, but each has its own **Hardware Configuration (HWC):** Primary or Secondary. The selected HWC is the HWC that you can go online with, download to, upload from, and so on.

Three types of basic CPU redundancy are available. These redundancy systems can be used in combination with Genius redundancy schemes.

- **Single Bus with Preferred Master (Series 90-70 CPUs CGR772 and CGR935 only):** uses a single Genius bus with one or more bus Controllers in each Controller. The primary unit is always chosen as the active unit when the units initially synchronize.

- **Single Bus with Floating Master (PACSystems RX7i, PACSystems RX3i, and Series 90-70):** uses a single Genius bus with one or more bus Controllers in each Controller. No switchover occurs on initial synchronization to make the primary unit the active unit.

- **Dual Bus with Floating Master (RX7i, RX3i, and Series 90-70):** uses dual busses with one or more bus Controllers in each Controller. No switchover occurs on initial synchronization. Bus Switching Modules (BSMs) are required in accordance with configuration of a dual bus network. This option provides redundancy of both the CPU and the Genius bus.
Genius Redundancy

A Genius redundancy system contains duplicate components that are configured to keep the Genius system operating properly even if one of the duplicate components fails or is taken out of service. Genius redundancy systems can be used in combination with PACSystems® RX7i, PACSystems® RX3i, and Series 90*-70 CPU redundancy systems.

You can configure five types of Genius redundancy systems:

- Genius dual bus redundancy (paired GBC internal)
- Genius dual bus redundancy (paired GBC external)
- Genius dual GBC redundancy (paired GBC internal)
- Genius dual GBC redundancy (paired GBC external)
- Genius dual bus & dual GBC redundancy

Hot CPU Redundancy Over Genius

A CPU Redundancy Over Genius system contains duplicate components that are configured to keep the system operating properly if one of the duplicate components fails or is taken out of service.

Five types of CPU Redundancy Over Genius systems can be built upon the various types of redundancy Genius systems combined with the basic CPU Redundancy schemes.

- CPU redundancy (GHS) using Genius dual GBC redundancy (paired GBC external): Single bus with preferred master (Series 90-70 only)
- CPU redundancy (HSB/GDB) using Genius dual GBC redundancy (paired GBC external): Single bus with floating master
- CPU redundancy (HSB/GDB) using Genius dual bus redundancy (paired GBC external): Single bus with floating master
- CPU redundancy (HSB/GDB) using Genius dual bus and dual GBC redundancy: Dual bus with floating master
- CPU redundancy (HSB/GDB) using a mixed Genius redundancy scheme

Want to know more? In the online help index, look up "Redundancy: an Overview".
Configuring Hot Redundancy Systems

The procedures below are the general steps to follow for configuring hot redundancy systems. For the procedures on configuring specific hot redundancy systems, see online help.

To set up the primary Hardware Configuration for hot CPU redundancy (PACSystems* RX7i, PACSystems* RX3i, and Series 90*-70)

1. In the Project tab of the Navigator, expand the target for which you want to set up CPU redundancy.

2. Right-click Hardware Configuration, point to Redundancy, and choose Wizard.

The Redundancy Wizards dialog box appears, with the Set up a Primary Hardware Configuration for CPU Redundancy option selected by default.
3. Click **Next** and follow the wizard to the end.

The wizard adds a redundancy CPU and other redundancy modules to the configuration.

The target property, **Dual HWC**, is now available in the Inspector and is set to False by default.

*To add Genius Bus Controllers (GBC) to your system (PACSystems* RX7i, PACSystems* RX3i, and Series 90*-70)*

1. In the **Project** tab of the **Navigator**, expand the target for which you want to set up Genius redundancy.

2. Right-click **Hardware Configuration**, point to **Redundancy**, and choose **Wizard**.

   The Redundancy Wizards dialog box appears.

3. Select **Add GBCs for Genius Redundancy**.
4. Click **Next** and follow the wizard to the end. The wizard enables you to select a Genius redundancy system and the location of the GBC modules. You can run this wizard multiple times to configure additional pairs of redundancy busses in the same system.

**Note:** When you add a Genius Bus Controller (GBC) to a PACSystems* RX7i, PACSystems* RX3i, or Series 90*-70 rack, a new Genius bus network is automatically created and associated with that slot and GBC module. Up to 31 Genius I/O devices (blocks) can be connected to a GBC through its Genius bus.

**To configure the secondary Hardware Configuration (RX7i, RX3i, and Series 90-70)**

1. In the **Project** tab of the **Navigator**, select the **Hardware Configuration**.

2. Right-click Hardware Configuration, point to **Redundancy**, and choose **Mirror to Secondary Hardware Configuration**.
A secondary rack system that is a copy of the primary rack system is generated. The target now displays two Hardware Configurations (HWCs), one labeled [Primary] and the other, [Secondary]. The Primary HWC is bolded because it is currently selected. The target property Dual HWC is now set to True. To select which Controller to interact with, set the Selected HWC property, just below the Dual HWC property.

**Note:** You can mirror as many times as necessary to synchronize the two HWCs after modifying the primary HWC. Each time you mirror the primary HWC, the secondary HWC is updated to reflect those changes.

---

Want to know more? In the online help index, look up "Redundancy: an Overview".
Hardware Configuration

DSM324i and Motion Mate DSM314 Motion Modules

DSM324I AND MOTION MATE DSM314 MOTION MODULES

The Series 90*-30 Controller family supports various I/O modules (discrete input, discrete output, discrete mixed, analog input, analog output, and analog mixed), communication modules, intelligent modules, bus Controllers, and motion modules. You configure Series 90-30 hardware as described in the section “Configuring Controller Hardware”, on page 61.

PACSystems’ RX3i and Series 90-30 support some Motion modules that no other GE IP Controller family supports: the DSM324i and the Motion Mate DSM314 (see page 153). Both are high performance, easy-to-use multi-axis motion control modules. Compatible with Controller logic solving and communications functions, the DSM314 supports the following servo types:
- Digital - GE digital servo amplifiers and motors.
- Analog - GE SL Series analog servos and third-party servos.

The DSM324i supports only the Digital servo types.

Both the DSM324i and the Motion Mate DSM314 modules have four axes that can be individually configured in Standard or Follower mode.

In order to perform motion programming in Logic Developer - PLC*, you must program a DSM324i or a Motion Mate DSM314 module (see page 153).

To add a DSM324i or Motion Mate DSM314 module

1. With a rack of the Hardware Configuration node expanded, right-click the empty slot you want to add a DSM324i or a Motion Mate DSM314 module to and choose Add Module.

   The Catalog dialog box appears.

2. In the Catalog dialog box, click the Motion tab.

   A selection of motion modules appears in the Catalog dialog box.

3. From the list, select DSM324i or Motion Mate DSM314.
Hardware Configuration

DSM324i and Motion Mate DSM314 Motion Modules

The following picture displays some of the selections available for a PACSystems® RX3i.

4. Click OK.

As per the example illustration, an IC693DSM314 is added to the Hardware Configuration of your project.

To configure a DSM324i or a Motion Mate DSM314

1. In the Project tab of the Navigator, double-click the slot containing a DSM324i or Motion Mate DSM314.

   The Parameter editor appears.

2. Configure the DSM324i or Motion Mate DSM314 by using the Parameter editor.

   Want to know more? In the online help index, look up “IC693DSM314: Motion Mate DSM314” and “IC693DSM324: DSM324i”.

GFK-1918R

Proficy® Logic Developer - PLC® 8.00
**Remote I/O**

**Series 90*-70 Genius Remote I/O Scanner**

A Remote I/O Scanner (IC697BEM733) is a Genius device that is part of a Genius Bus system. As such, it is added as a Genius device to the Genius Bus Controller (GBC) and it is represented in Logic Developer - PLC* by an SBA node under the GBC in a Series 90*-70 target, as illustrated to the left.

It is also represented by a separate target in the Machine Edition* project, as illustrated below.

A Remote I/O Scanner resides in Slot 1 of the Series 90-70 main rack. I/O modules can be inserted into the other slots of the main rack and on all the slots of the expansion racks. As such, the Remote I/O Scanner is represented by a target in the Project tab of the Navigator, and is also represented by a module in slot 1 of the main rack of that target.
To add a Remote I/O Scanner

1. In the Project tab of the Navigator, expand the Hardware Configuration and expand the main rack.

2. Right-click the slot containing the Genius Bus Controller (GBC) and choose Add Genius Device.

   The Choose SBA dialog box appears.

3. Select a Serial Bus Address (SBA) for the Genius device and click OK.

   The Catalog appears.

4. Select the device you want to add and click OK.

   The device is added to the bus configuration with the SBA you selected.

5. When prompted to add a new target for the Remote I/O Scanner, click Yes.

   A new GE Remote I/O target appears in your project. This target represents the Remote I/O Scanner. In the Hardware Configuration of the new target, the Remote I/O Scanner also appears in Slot 1 of the main rack.

6. Configure the communication with the Remote I/O Scanner target.

7. Add and configure I/O modules for the rack system of the Remote I/O Scanner target.

8. Configure the Remote I/O Scanner module’s parameters in Slot 1 of the main rack of the Remote I/O Scanner target.

9. Set up the Remote I/O Scanner Genius device in the same SBA node.

   The I/O Map of both Remote I/O Scanner configurations must match (at the Remote I/O Scanner target level and at the Remote I/O Scanner Genius Bus device level). Now whenever you update the I/O map on the Remote I/O Scanner, the I/O map is automatically updated to the GBC.

10. Download the Hardware Configuration of the new Series 90*-70 Remote I/O target.

11. Download the Hardware Configuration of the original Controller target that contains the GBC.

Want to know more? In the online help index, look up “IC697BEM733: Remote I/O Scanner.”
**VersaMax** Remote I/O

VersaMax remote I/O consists of a Network Interface Unit (NIU) (Ethernet, Genius, or Profibus), and one or more I/O modules. You can use Logic Developer - PLC to configure this hardware. Once configured, the remote I/O can be controlled by a VersaMax Controller or a PC Controller. The differentiating factor between a VersaMax remote I/O and a Controller is that a remote I/O is simply an input/output device with a communication interface. Unlike a Controller, a remote I/O has no CPU.

With Logic Developer - PLC, you can add racks, configure the power supply, and configure modules in the VersaMax Remote I/O Hardware Configuration.

Remote I/O targets are generally added to a project when you create a project from a template. You can also add a GE Remote I/O target to an existing project.

**To create a project containing a GE Remote I/O target from a template**

1. From the **File** menu, choose **New Project**.
2. From the Project Template list, choose the GE Remote I/O you want to add to your project.
3. Enter a descriptive **Project Name**.
4. Click **OK**.

A new Remote I/O project is started.

**To add a GE ip Remote I/O target to an existing project**

1. In the **Project** tab of the **Navigator**, right-click the **Project node**.
2. Point to **Add Target**, then to **GE Intelligent Platforms Remote I/O**, and choose a remote I/O.
**Note:** The **Series 90-70 Genius** option is a Series 90*-70 Remote I/O Scanner (see page 78).

**To replace the power supply in your remote I/O configuration**

1. In the Project tab of the Navigator, expand the Hardware Configuration of the remote I/O.

   The Navigator displays the following:

   ```
   <Project Tree Display>:
   VersaMax Remote
   Target
   - Data Watch Lists
   - Hardware Configuration
   - Main Rack
   - PWR (IC695PWR001)
   - Slot 0 (IC695EM001)
   ```

2. Right-click the PWR slot and choose Replace Module.

   The Module Catalog dialog box appears:

   ```
   Module Catalog
   Power Supply
   ```

<table>
<thead>
<tr>
<th>Catalog Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC695PWR002</td>
<td>Power Supply with Expandable 1/2/4/8 DC Input</td>
</tr>
<tr>
<td>IC695PWR101</td>
<td>Power Supply 1/2/4/8/16/32 DC Input</td>
</tr>
<tr>
<td>IC695PWR201</td>
<td>Power Supply 2/4 DC Input</td>
</tr>
<tr>
<td>IC695PWR202</td>
<td>Power Supply with Expandable 1/2/4/8 DC Input</td>
</tr>
</tbody>
</table>

3. From the list, select the power supply that you want to configure for your system.

4. Click OK.

**To add a new Carrier/Base to your VersaMax* Remote I/O**

1. In the Project tab of the Navigator, select a remote I/O target.

2. Expand the Hardware Configuration node and the main rack.
Hardware Configuration
Remote I/O

The Navigator displays the following:

3. Right-click the Slot 0 node and choose Add Carrier/Base.

The Module Catalog dialog box appears.

4. Select the carrier/base that you want to add to the remote I/O target.
   
   Note: You can add a maximum of eight carrier modules to each VersaMax* rack.

5. Click OK.

   To add a module to a carrier/base

   1. In the Project tab of the Navigator, expand the Remote I/O target you want to add a base to.

   2. Double-click an empty carrier/base.
The Module Catalog appears.

3. From the Module Catalog, select the module that you want to install.
4. Click **OK**.

*Want to know more?* In the online help index, look up “Hardware Configuration”.
Hardware Configuration

Remote I/O
Program Types

For most GE IP Controllers, all the logic that is downloaded to a Controller constitutes one program. However, there are cases when distinctions need to be made. Therefore, we can speak of a main program, a C program, and a Motion program.

Each C program is a named section of executable code. The main program and the Motion program contain blocks; that is, named sections of executable code that can be written in various languages.
Logic Programs and Blocks

The Project tab of the Navigator contains all of the programs:
- The main program and its blocks reside in the Program Blocks folder. You can create user-defined subfolders to further organize these blocks.
- C programs reside in the Logic folder.
- The Motion program resides in the Motion Program folder.

Main Program

Most GE IP Controllers support only a main program. The main program consists of a block of logic named _MAIN and optionally, one or more of the following kinds of blocks of logic, found under the Program Blocks folder in the Logic folder of a GE IP Controller target:
- Called blocks developed in various languages:
  - Ladder Diagram (LD) Blocks
  - Structured Text (ST) Blocks
  - Function Block Diagram (FBD) Blocks
  - Instruction List (IL) Blocks
  - C Blocks
- Called parameterized blocks (LD, ST, FBD, and/or C).
- Interrupt blocks (LD, ST, FBD, C, and/or IL) scheduled to execute at certain time intervals or when specific memory registers reach certain values.
- User-defined function blocks (UDFBs) in LD, ST, and FBD, of which independent instances with their own data structures and local memory can be used in logic.

Want to know more? In the online help index, look up "UDFB" then choose "User-Defined Function Blocks (UDFBs) (PACSystems)".

Specialty function blocks, like HART utilities. Like UDFBs, you can create independent instances of specialty functions blocks with their own instance data (a structure variable). Unlike UDFBs, you cannot edit the logic of specialty function blocks because they are read-only and provided to you by GE.

Want to know more? In the online help index, look up "Specialty Function Blocks (PACSystems)".

All GE IP Controllers support LD blocks, but only certain Controller families or models support one or more of the other kinds of blocks.
The _MAIN block can be in various languages, depending on the Controller family or model.

You download the main program to the Controller as logic; that is, after you initiate a download, a dialog box presents you with various Download to Controller options and you select the Logic option. If the Logic option is not present, select the Program option.

**C Programs**

Series 90*-70 CPUs firmware version 6.00 and later support either the main program, or various C programs, or a combination of a main program and C programs. You download these various programs to the Controller together as logic; that is, after you initiate a download, a dialog box presents you with three Download to RAM options and you select the Logic option.

If there is no _MAIN block, then there is no main program. In this case, you would have only one or more C programs and possibly various standalone interrupt blocks (LD or C). These standalone interrupt blocks do not constitute a main program if there is no _MAIN block.

A C program can be named _MAIN only if the target contains no LD blocks.

A C program is not the same as a C block. For more information, see “C Programs”, on page 137.

**Motion Program**

In addition to the main program, Series 90*-30 CPUs firmware version 10.00 and later, as well as PACSystems* RX3i CPUs firmware version 2.80 and later, support a Motion program for use on a DSM324i or Motion Mate DSM314 module. The Motion program consists of:

- CAM profiles
- CAM blocks
- Local Logic block
- Motion blocks

You download the Motion program to the Controller as part of its Hardware Configuration (HWC); that is, after you initiate a download, a dialog box presents you with various Download to RAM options and you select the Hardware Configuration option. If you want to download the main program at the same time, you also select the Logic option. If neither option is present, select the
Program option to download both the Hardware Configuration and the Logic. For more information about the Motion Program, see page 153.

**Number of Blocks in the Main Program**

<table>
<thead>
<tr>
<th>CPU Type</th>
<th>Number of Blocks</th>
<th>Maximum Number of Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>PACSystems*</td>
<td>511 subroutine blocks plus one _MAIN block</td>
<td>512</td>
</tr>
<tr>
<td>Series 90*-70</td>
<td>255 subroutine blocks plus one _MAIN block</td>
<td>256</td>
</tr>
<tr>
<td>Series 90*-30</td>
<td>64 subroutine blocks plus one _MAIN block</td>
<td>65</td>
</tr>
<tr>
<td>VersaMax* and VersaMax* Micro</td>
<td>64 subroutine blocks plus one _MAIN block</td>
<td>65</td>
</tr>
<tr>
<td>VersaMax* Nano</td>
<td>8 subroutine blocks plus one _MAIN block</td>
<td>9</td>
</tr>
<tr>
<td>Series 90* Micro</td>
<td>1 _MAIN block</td>
<td>1</td>
</tr>
</tbody>
</table>

**Scheduling Programs**

On GE IP Controllers that support only one program (the main program), the program as a whole cannot be scheduled. Every scan, logic execution begins with the _MAIN block. You can schedule individual interrupt blocks of logic but not the _MAIN block.

When you execute both a main program and a Motion program on a Series 90*-30 Controller with a CPU firmware version 10.00 and later, or on a PACSystems* RX3i with a firmware version of 2.80 or later, you cannot schedule the programs because they are executed concurrently in two different locations.

- In the CPU, every scan, logic execution begins with the _MAIN block of the main program.
- In the DSM324i or Motion Mate DSM314 module, the Motion program executes independently of CPU scan times.
- You can schedule individual interrupt blocks of logic in the main program (except the _MAIN block).

Series 90*-70 CPUs firmware version 6.00 and later support scheduling programs. On targets that contain such CPUs, you can schedule any program, even the main program. That is, logic execution of any scan does not have to begin with the _MAIN block of the main program: you can schedule any C program to begin the logic execution part of the scan. Within the main program,
you can further schedule individual interrupt blocks of logic (except the _MAIN block). If you have no main program, that is, if your logic has no _MAIN block, you can still have stand-alone interrupt blocks that you can schedule individually.

To schedule the execution of programs on a Series 90*-70

1. In the Project tab of the Navigator, right-click a C program or the _MAIN LD block, and choose Properties.
   The Inspector displays the program properties. The main program properties (Program Name, Scheduling, and Stack Size) are shown together with the _MAIN block’s individual block properties.
2. In the Inspector, expand the Scheduling property.
3. Select a Schedule Mode.
4. Select the other properties used in the schedule mode you selected.
5. Click OK.
   Note: If you made any entries, they are validated when you click OK. You must correct any errors before you can click OK to close the dialog box.

To create a user-defined folder

1. In the Project tab of the Navigator, expand the target with the main program you want to organize and then expand the Logic folder.
2. Right-click the Program Blocks folder, point to New, and choose Folder.
   A new user-defined folder appears with a default name.
3. Optionally enter a new folder name, which must be unique among the folders directly under the parent folder.

To schedule the execution of a block of logic

1. In the Project tab of the Navigator, right-click an existing LD, FBD, ST, C, or IL block, and choose Properties.
   The block’s properties appear in the Inspector.
2. In the Inspector, click the button in the Scheduling property.

Want to know more? In the online help index, look up "Programs on GE IP Controllers: an Overview".

Want to know more? In the online help index, look up "organizing logic (main program of a GE IP Controller)".
Logic Programs and Blocks

The Scheduling dialog box appears.

3. Configure scheduling by entering values in each of the fields.

Want to know more? In the online help index, look up “Scheduling execution of blocks of logic”.

To control access to a block

1. Right-click an LD, FBD, ST, or IL block and choose Properties.
2. In the Inspector, expand the Lock Settings property.
3. In the Lock Type property, choose a setting from the list.
4. In the Password property, enter a password.

To search/replace in one block

1. Double-click an LD, FBD, ST, or IL block to open it.
2. From the Search menu, choose Find or Replace.
   A dialog box appears.
3. In the Find what or in the Text to find field, enter the text to find.
4. (Only if you want to replace text.) In the Replace with field or in the New text field, enter the text to replace the found text with.
5. Select or clear each search/replace option, as required.
6. Click Find, Find next, Replace, Replace All, Close, or Cancel, as required.
Note: With Logic Developer - PLC®, you can search for some text in an entire target or portions thereof, which you can specify. Various options are available to narrow a search.

Want to know more? In the online help index, look up "Global" and select the "Conducting a Global Search" help topic.

Indirect References

The LD, FBD, and ST editors support indirect references. An indirect reference treats the value of a variable assigned to an instruction operand as a pointer to other data, rather than as actual data. Indirect references are sometimes referred to as relative pointers.

Indirect references can be used only as follows:
- With PACSystems* CPUs and Series 90*-70 CPUs.
- In the LD, FBD, and ST editors.
- With %R, %AI, %AQ, %P, %L, and %W memory areas.
- On instructions that support indirect references.

Notes for PACSystems*
- The index for an indirect reference to %W is a 32-bit DWORD value.
- Indirect references cannot be used to address bits in 16-bit memory.
- Indirect references are not supported on symbolic variables or I/O variables.

Possible Uses
- To perform the same operation to many registers.
- To avoid repetitious logic within the application program.
- In loop situations where each register is incremented by a constant or by a value specified until a maximum is reached.

To assign an indirect reference
- Where a variable operand is expected, type the @ symbol, followed by a valid reference address or variable name.
  The LD, FBD, or ST editor converts a reference address to a variable name or the editor substitutes the name of a variable already mapped to the address.
Notes:

- The actual variable name does not contain the @ symbol. The @ symbol, when used at the start of a reference address or variable name, merely indicates that the address or variable must be treated as an indirect reference.

- The @ symbol can be thought of as an operator with the lowest precedence. In other words, MyArray[2,5] becomes a reference address that is used for the indirect reference. Another way to think of this is @\(\text{MyArray[2,5]}\) as opposed to \((@\text{MyArray})[2,5]\).

The following are valid indirect references:

- @R0001 (a variable name automatically created from a reference address)
- @MyArray[2,5] (an array reference)
- @$MainSwitch (a universal variable)
- @MyTimer.PV (a structure element)
The Ladder Diagram (LD) editor is used to create programs with the Ladder Diagram programming language. LD logic graphically represents the programmed actions performed by a Controller as it executes.

The LD editor is cell-based with rungs constructed of horizontal sequences of instructions that are wired together. A given instruction and its operands can occupy one or more cells.

You can work with the LD editor while offline to edit a disk copy of a project, or while online to monitor the execution of the logic while you fine tune the project, for example by making word-for-word changes (see page 102).

You can customize the appearance and behavior of the LD editor by setting options (see page 26).

An LD block is a named section of LD Logic that is compiled and downloaded to the Controller represented by the associated target.

To customize the LD editor

1. In the Options tab of the Navigator, expand the Editors folder and then the Ladder folder.
2. Right-click a page (Confirmations, Editing, Font and Colors, or View), and choose Properties.
   The configurable options appear as properties in the Inspector.
3. In the Inspector, adjust settings as required.

To create an LD block

1. In the Project tab of the Navigator, right-click Program Blocks, point to New, and then choose LD Block.
   A new LD block with a default name is created.
2. Rename the block as desired.

To open an LD block for editing

- In the Project tab of the Navigator, double-click an LD block.
  The block opens in the LD editor.

Note: You can have multiple blocks open for editing. To navigate to another open LD block, click the tab that displays its name at the top of the editor window.
Working with the LD Editor Offline

When you are offline, there is no live communication between the LD editor and the target. Most logic development is done while offline. The following diagram illustrates some of the more common operations you can perform when you use the LD editor offline.

- Drag variables to instruction operands to assign them.
- Right-click to insert an instruction.
- Use the Inspector to configure address and length for instructions.
- The Companion automatically provides information on what you click.
- Insert common instructions and wiring with the toolbar.
- Drag LD instructions from the Toolchest.
- Displays a list of the places the selected variable is used.
- In place editor. Comment runs.

Insert common instructions and wiring with the toolbar.
Drag variables to instruction operands to assign them.
Right-click to insert an instruction.
Use the Inspector to configure address and length for instructions.
The Companion automatically provides information on what you click.
To insert an instruction

1. In the LD editor, right-click an empty cell and choose Place Instruction. A smart list appears listing all available instruction mnemonics.
2. Choose an instruction mnemonic from the list and press ENTER.
   
   Note: If you attempt to place an instruction that has inputs into the first column, it is pushed to the second column and a horizontal wire is inserted in the first column.

To assign instance data to a built-in function block instance and assign a length to an instruction

There are two methods.

First method

1. In the LD editor, insert a built-in function block or insert an instruction that requires a Length.

   In the following example, a built-in function block instance has a length. Most built-in function blocks do not require their instances to have a length.

   ![Diagram of a built-in function block instance]

   Four question marks indicate that the Address property of the function block instance has not been initialized. As a result, no instance data has been assigned to the instance.

   “1” is the default value of the Length property of an instruction.

2. Right-click the function block instance or instruction and choose Properties.

   The Inspector displays the instance’s or instruction’s properties.

   ![Inspector showing Address and Length properties]

3. In the Address property, enter a variable name or a reference address to specify the start of a memory block used for the instance data of the function block instance.

   Note: If you type a reference address, it is converted to a variable name automatically.
4. In the Length property, enter the number of bits or 16-bit registers on which the function block instance or instruction is to operate.

**Second method**

1. In the LD editor, insert a built-in function block or insert an instruction that requires a Length.

2. Double-click the function block instance or instruction or select it and press ENTER.

   When you double-click a function block instance, the Function Block Properties dialog box appears, as pictured below. When you double-click an instruction, the Function Properties dialog box appears, in which the Address property is unavailable.

   ![Function Block Properties](image)

3. In the Address property, enter a variable name or a reference address to specify the start of a memory block used for the instance data of the function block instance.

   **Note:** If you type a reference address, it is converted to a variable name automatically.

4. In the Length property, enter the number of bit or 16-bit registers on which the function block instance or instruction is to operate.

**To assign variables to instruction operands**

1. In the LD editor, double-click beside any operand of an instruction, or click there and press ENTER.

   A smart list appears.

2. Type, or choose from the list, a variable name or reference address.

   If you enter a reference address, a variable name is automatically substituted.

**To check (validate) a single LD block**

1. In the Project tab of the Navigator, expand the target that contains the block to check, and then expand the Logic folder.

2. Expand the Program Blocks folder and then, if the block resides in a user-defined folder, expand the user-defined folder.
3. Right-click the block and choose **Check Block**.

Machine Edition* checks the block for errors. Any messages or errors are displayed in the Build tab of the Feedback Zone.

**Editing Logic as Text**

You can enter LD logic into the LD editor by starting to type right into a cell. A smart list appears and you can continue to type your instructions and operands. When you press ENTER, the logic you typed appears in customary graphic form in the LD editor. For example, in the first cell of a new rung, if you type “AD MyDint1 MyDint2 Result12;AD MyDint3 MyDint4 Result34”, the following logic appears in the LD editor:

```
[Diagram showing LD logic with instructions and operands]
```

A horizontal wire was inserted in the first cell to make room for the ADD_DINT instruction's operands and the operands were inserted in their proper places. In the keyboard entry, “AD” stood for “ADD_DINT”, because ADD_DINT is the first available instruction that begins with “AD” in the smart list. A semicolon separates the ADD_DINT instructions. The required horizontal wires are supplied automatically.

**Want to know more?** In the online help index, look up "Keyboard Method to insert LD instructions".

A powerful extension of typing logic into the LD editor is the ability to write LD logic in any text editor and copy and paste it into the LD editor, or to copy and paste logic from the LD editor to a text editor. You can copy an entire block of LD logic from the Navigator to any text editor, or copy sections of LD logic as text from the LD editor to any text editor. You can then edit the logic in the text editor and copy the edited logic back into the LD editor.

When LD logic is copied as text, a verbose format is used to make it easier to read, but when you edit the logic as text you can use the same shorthand used in the keyboard method of entering LD logic in the LD editor.

**To copy an entire LD block as text**

1. In the `Project` tab of the `Navigator`, expand the target and expand the `Logic` folder.
2. Expand the Program Blocks folder and, if the LD block resides in a user-defined folder, expand the user-defined folder.

3. Optionally right-click the LD block and choose **Check Block**. This validates the LD block. If the logic contains errors, error messages appear in the Feedback Zone. If the logic contains errors, fix them.

4. Right-click the LD block and choose **Copy**.
   The contents of the LD block are copied to the Windows Clipboard.

5. Paste the content of the Windows Clipboard into a text editor.
   The LD logic is pasted as text.

**To copy a section of LD logic as text**

1. Optionally right-click the LD block and choose **Check Block**. This validates the LD block. If the logic contains errors, error messages appear in the Feedback Zone. If the logic contains errors, fix them.

2. In the LD editor, select the cells that contain the logic you want to copy as text.

3. Right-click the selected logic and choose **Copy**.
   The LD logic is copied to the Windows Clipboard.

4. Paste the content of the Windows Clipboard into a text editor.
   The LD logic is pasted as text.

**To copy text into the LD editor**

1. In the text editor, select and copy the text representing LD logic to the Windows clipboard.

2. In the LD editor, do one of the following:
   - Select the cells that you want to overwrite.
   - or -
   - Click the cell that will be at the upper left corner of the LD logic you want to paste.

3. Right-click and choose **Paste**.
   Existing cells in the LD editor are overwritten with the content of the Windows clipboard.

**To move or duplicate LD logic**

1. In the LD editor, select a range of logic.

2. To move it:
   - Click the selection and drag it to a new location.
To duplicate it:

- Press CTRL while clicking the selection, and drag the selection to where you want to place the duplicate.

When you release the mouse button, the selection is respectively moved to the new location or a duplicate copy of the selected logic is placed in the new area.

**Want to know more?** In the online help index, look up “Copying LD Logic as Text: Overview”.

**Working with the LD Editor Online**

When online in monitor mode, you can monitor the Controller but cannot change anything in the Controller or edit your logic. The LD editor animates LD logic to reflect program execution in the Controller. Data values change typically every 250 milliseconds while coils and contacts indicate power flow. The number of milliseconds is set in the Update Rate target property. (See “To configure a Controller target”, on page 35.)

When you are online in programmer mode, you can edit your logic, make changes in the Controller, and monitor the Controller. Many GE targets also support Run Mode Store; that is, you can download logic to them when they are running.

See “Offline, Online: monitor mode, programmer mode”, on page 44.
Three methods are at your disposal to make changes in your LD logic and download the changes to an online running Controller:

- **Test Edit** *(PACSystems* only)*: see page 100.
- **Word-for-word changes**: see page 102.
- **Go not equal, keep working, and download changes**: see page 103.

You can also affect BOOL variables during execution. See page 103.

**Test Edit (PACSystems* only)**

With Test Edit, you can perform transacted online programming. That is, you can modify an LD block in the LD editor while seeing both the original and modified
logic, test the new logic's execution on the PACSystems, and then either keep
the modified logic or roll back to the original logic.

To begin editing in Test Edit
1. In the LD editor, start modifying the logic.
   The Logic Change Options dialog box appears.
2. Select Enter Test Edit Mode and click OK.
   The LD editor is placed in Online Edit mode, with some visual cues to remind
   you that a Test Edit session is in progress. Because Test Edit is an extension of
   the Run Mode Store (RMS), only changes supported for an RMS are supported
   in Test Edit mode. At any time, you can test the modified logic.

To begin testing the modified logic
1. Click anywhere inside the LD editor.
2. From the Debug menu, choose Begin Test.
   The modified logic is downloaded to the PACSystems, which then executes the
   modified logic. The original block of logic is retained in PACSystems memory as
   an inactive block.
   You now have three possible courses of action:
   ▪ Cancel the test and continue editing the logic while in Test Edit mode.
   ▪ Cancel Test Edit mode and restore the original logic in the PACSystems.
   ▪ Accept the changes you tested and commit them to the PACSystems.

To cancel the test and continue editing the logic while in Test Edit mode
1. Click anywhere inside the LD editor.
2. From the Debug menu, choose Cancel Test.
   The modified logic block is deactivated in the PACSystems*. The original block
   is activated and executed. The modified logic block retains its changes in the LD
   editor and you can continue editing it and test it again later.

To cancel Test Edit mode and restore the original logic in the PACSystems
1. Click anywhere inside the LD editor.
2. From the Debug menu, choose Cancel Edit.
   If you made no changes to logic, the Test Edit session ends immediately. If you
   made any changes to logic, the Cancel Test Edit dialog box appears.
3. In the Cancel Test Edit dialog box, select one of the following options and click
   OK.
Logic Programs and Blocks

LD Editor

- **Restore original logic in editor and maintain equality:** The PACSystems deactivates and deletes the modified block of logic, and it activates and executes the original block. The LD editor discards the modified logic from the block, retaining only the block's original logic. Your Test Edit session ends. You are online in programmer mode and logic equal.

- **Keep modified logic in editor and lose equality:** The PACSystems deactivates and deletes the modified block of logic, and it activates and executes the original block. The LD editor retains the modified block of logic and no longer displays the original logic where it was different. Your Test Edit session ends. You are online in programmer mode and logic not equal.

To accept the changes you tested and commit them to the PACSystems*

1. Click anywhere inside the LD editor.
2. From the Debug menu, choose **Accept Edit**.

The modified logic is committed to the PACSystems. The original logic is completely deleted from the PACSystems; it no longer exists as an inactive block. The Test Edit session is ended. You are online in programmer mode and logic equal.

**Word-for-Word Changes**

A word-for-word is a small change in logic made while online that generally fits in the same amount of memory as the original logic. For example, changing the type of contact or coil or changing an operand is usually a word-for-word change.

Word-for-word changes can be completed online to PACSystems*, Series 90*-70, Series 90*-30, VersaMax, and Series 90 Micro Controllers. (VersaMax Nano / Micro Controllers do not support word-for-word changes.)

To make word-for-word changes

1. While online to a target Controller, make a change to LD logic that does not change the logic size.
2. Do one of the following:
   - If the Logic Change Options dialog box appears, select Word-for-word change and click **OK**.
   - or -
   - If the Word for Word Change dialog box appears, click **Yes**.

The change is downloaded to the Controller.
Go not equal, keep working, and download changes

If you make changes in logic that do not qualify for Test Edit or a word-for-word change, or if you choose not to enter Test Edit mode or make a word-for-word change, logic becomes not equal.

To regain equality, you can download your changes to the Controller (see page 47) or upload the logic from the Controller (see page 48). You can generally download to the Controller whether the target is running or not. All Controller families support this Run Mode Store capability, but not every Controller in every family.

Affecting BOOL variables

To turn on/off or force a variable

- In the LD editor, right-click a BOOL variable anywhere in LD logic and choose Force ON, Force OFF, Toggle Force, Turn ON, Turn OFF, or Toggle IO.

Note: Forcing a variable ON or OFF overrides any actions the application may take during runtime. That is, if a variable is forced OFF (0), but LD logic is trying to set it to ON (1), it remains set to OFF.

Want to know more? In the online help index, look up “Test Edit: an Overview”, “working online with variables and LD logic: overview”, and “Run Mode Store (RMS)”.  

LD Instructions

The following is a list of all LD instructions available in Logic Developer - PLC*.

Companion help indicates which Controllers support the instructions.

Want to know more? In the online help index, look up “LD Instructions.”
Advanced Math

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACOS</td>
<td>Arc cosine</td>
</tr>
<tr>
<td>COS</td>
<td>Cosine</td>
</tr>
<tr>
<td>SIN</td>
<td>Sine</td>
</tr>
<tr>
<td>TAN</td>
<td>Tangent</td>
</tr>
<tr>
<td>LN_LREAL^P</td>
<td>Natural logarithm of x</td>
</tr>
<tr>
<td>LN_REAL^P</td>
<td>Natural logarithm of x</td>
</tr>
<tr>
<td>EXP_LREAL^P</td>
<td>Exponential of x</td>
</tr>
<tr>
<td>EXP_REAL^P</td>
<td>Exponential of x</td>
</tr>
<tr>
<td>LOG_LREAL^P</td>
<td>Logarithm of x</td>
</tr>
<tr>
<td>LOG_REAL^P</td>
<td>Logarithm of x</td>
</tr>
<tr>
<td>SORT_DINT</td>
<td>Sort integer</td>
</tr>
<tr>
<td>SQRT_LREAL^P</td>
<td>Square root of x</td>
</tr>
<tr>
<td>SQRT_REAL</td>
<td>Square root of x</td>
</tr>
<tr>
<td>ACOS_LREAL^P</td>
<td>Arc cosine of real x</td>
</tr>
<tr>
<td>COS_LREAL^P</td>
<td>Cosine of real x</td>
</tr>
<tr>
<td>SIN_LREAL^P</td>
<td>Sine of real x</td>
</tr>
<tr>
<td>TAN_LREAL^P</td>
<td>Tangent of real x</td>
</tr>
<tr>
<td>ACOS_REAL^P</td>
<td>Arc cosine of real x</td>
</tr>
<tr>
<td>COS_REAL^P</td>
<td>Cosine of real x</td>
</tr>
<tr>
<td>SIN_REAL^P</td>
<td>Sine of real x</td>
</tr>
<tr>
<td>TAN_REAL^P</td>
<td>Tangent of real x</td>
</tr>
<tr>
<td>ASIN</td>
<td>Arc sine</td>
</tr>
<tr>
<td>ASIN_LREAL^P</td>
<td>Arc sine of real x</td>
</tr>
<tr>
<td>ASIN_REAL^P</td>
<td>Arc sine of real x</td>
</tr>
<tr>
<td>DEG_TO_RAD_REAL^P</td>
<td>Degrees to radians of x</td>
</tr>
<tr>
<td>DEG_TO_RAD_REAL^P</td>
<td>Degrees to radians of x</td>
</tr>
</tbody>
</table>

Bit Operations

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AND_DWORD^</td>
<td>Logical AND of DWORD</td>
</tr>
<tr>
<td>BIT_SET_DWORD^</td>
<td>Bit set of DWORD</td>
</tr>
<tr>
<td>NOT_DWORD^</td>
<td>Logical NOT of DWORD</td>
</tr>
<tr>
<td>OR_DWORD^</td>
<td>Logical OR of DWORD</td>
</tr>
<tr>
<td>SHIFTL_DWORD^</td>
<td>Shift left of DWORD</td>
</tr>
<tr>
<td>SHIFTL_WORD</td>
<td>Shift left of WORD</td>
</tr>
<tr>
<td>OR_WORD</td>
<td>Logical OR of WORD</td>
</tr>
<tr>
<td>SHIFTR_DWORD^</td>
<td>Shift right of DWORD</td>
</tr>
<tr>
<td>SHIFTR_WORD</td>
<td>Shift right of WORD</td>
</tr>
<tr>
<td>BIT_CLR_DWORD^</td>
<td>Bit clear of DWORD</td>
</tr>
<tr>
<td>BIT_CLR_WORD</td>
<td>Bit clear of WORD</td>
</tr>
<tr>
<td>BIT_TEST_DWORD^</td>
<td>Bit test of DWORD</td>
</tr>
<tr>
<td>BIT_TEST_WORD</td>
<td>Bit test of WORD</td>
</tr>
<tr>
<td>MASK_COMP_DWORD</td>
<td>Mask and compute of DWORD</td>
</tr>
<tr>
<td>MASK_COMP_WORD</td>
<td>Mask and compute of WORD</td>
</tr>
<tr>
<td>ROL_DWORD^</td>
<td>Rotate left of DWORD</td>
</tr>
<tr>
<td>ROL_WORD</td>
<td>Rotate left of WORD</td>
</tr>
<tr>
<td>XOR_DWORD^</td>
<td>Exclusive OR of DWORD</td>
</tr>
<tr>
<td>XOR_WORD</td>
<td>Exclusive OR of WORD</td>
</tr>
<tr>
<td>BIT_POS_DWORD^</td>
<td>Bit position of DWORD</td>
</tr>
<tr>
<td>BIT_POS_WORD</td>
<td>Bit position of WORD</td>
</tr>
<tr>
<td>MASK_COMP_DWORD</td>
<td>Mask and compute of DWORD</td>
</tr>
<tr>
<td>MASK_COMP_WORD</td>
<td>Mask and compute of WORD</td>
</tr>
<tr>
<td>ROR_DWORD^</td>
<td>Rotate right of DWORD</td>
</tr>
<tr>
<td>ROR_WORD</td>
<td>Rotate right of WORD</td>
</tr>
<tr>
<td>BIT_SEQ</td>
<td>Bit sequence</td>
</tr>
<tr>
<td>NOT_DWORD^</td>
<td>Logical NOT of DWORD</td>
</tr>
<tr>
<td>ROR_DWORD^</td>
<td>Rotate right of DWORD</td>
</tr>
<tr>
<td>ROR_WORD</td>
<td>Rotate right of WORD</td>
</tr>
</tbody>
</table>

Coils

<table>
<thead>
<tr>
<th>Coil Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COIL</td>
<td>Coil</td>
</tr>
<tr>
<td>NCCOIL</td>
<td>Not coil</td>
</tr>
<tr>
<td>NTCOIL^P</td>
<td>Not transistor coil^P</td>
</tr>
<tr>
<td>PTCOIL^P</td>
<td>Power transistor coil^P</td>
</tr>
<tr>
<td>SETCOIL</td>
<td>Set coil</td>
</tr>
<tr>
<td>CONTCOIL</td>
<td>Contact coil</td>
</tr>
<tr>
<td>NEGCOIL</td>
<td>Negative coil</td>
</tr>
<tr>
<td>POSCOIL</td>
<td>Positive coil</td>
</tr>
<tr>
<td>RESETCOIL</td>
<td>Reset coil</td>
</tr>
</tbody>
</table>

Communication

- MODBUS_TCP_RW (VersaMax Micro CPUs with firmware version 4.00 or later)
- PNIO_DEV_COMM

Contacts

<table>
<thead>
<tr>
<th>Contact Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONTCON</td>
<td>Contact</td>
</tr>
<tr>
<td>HIALR^</td>
<td>High alarm</td>
</tr>
<tr>
<td>NCCON</td>
<td>Not contact</td>
</tr>
<tr>
<td>NOCON</td>
<td>Not open contact</td>
</tr>
<tr>
<td>NTCON^P</td>
<td>Not transistor contact^P</td>
</tr>
<tr>
<td>PTCON^P</td>
<td>Power transistor contact^P</td>
</tr>
<tr>
<td>FAULT^</td>
<td>Fault</td>
</tr>
<tr>
<td>LOALR^</td>
<td>Low alarm</td>
</tr>
<tr>
<td>NEGCON^</td>
<td>Negative contact</td>
</tr>
<tr>
<td>NOFLT^</td>
<td>Not open fault</td>
</tr>
<tr>
<td>POSCON^</td>
<td>Positive contact</td>
</tr>
</tbody>
</table>

Control

<table>
<thead>
<tr>
<th>Control Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DO_IO</td>
<td>Digital output</td>
</tr>
<tr>
<td>EXIT_FOR^</td>
<td>Exit for</td>
</tr>
<tr>
<td>F_TRIG^P</td>
<td>Forward trig</td>
</tr>
<tr>
<td>PID_IND</td>
<td>Proportional integral control indicator</td>
</tr>
<tr>
<td>PID_IAS</td>
<td>Proportional integral control indicator</td>
</tr>
<tr>
<td>R_TRIG^P</td>
<td>Reverse trig</td>
</tr>
<tr>
<td>SCAN_SET_IO^P</td>
<td>Scan set io</td>
</tr>
<tr>
<td>SER</td>
<td>Serial</td>
</tr>
<tr>
<td>SVC_REQ</td>
<td>Service request</td>
</tr>
<tr>
<td>SUS_IO^</td>
<td>Susceptible io</td>
</tr>
<tr>
<td>SUSP_IO_INTR</td>
<td>Suspended io intr</td>
</tr>
<tr>
<td>SWITCH_POS^P</td>
<td>Switch position</td>
</tr>
</tbody>
</table>

^P indicates instructions exclusive to PACSystems Controllers.
^70 indicates instructions exclusive to Series 90-70 Controllers.
^ indicates instructions shared by PACSystems and Series 90-70 Controllers and exclusive to them.
### Conversions

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCD4_TO_INT</td>
<td>DINT_TO_INT</td>
</tr>
<tr>
<td>BCD4_TO_REAL</td>
<td>DINT_TO_LREAL</td>
</tr>
<tr>
<td>BCD4_TO_UINT</td>
<td>DINT_TO_REAL</td>
</tr>
<tr>
<td>BCD8_TO_DINT</td>
<td>DINT_TO_UINT</td>
</tr>
<tr>
<td>BCD8_TO_REAL</td>
<td>INT_TO_BCD4</td>
</tr>
<tr>
<td>DEG_TO_RAD</td>
<td>INT_TO_DINT</td>
</tr>
<tr>
<td>DINT_TO_BCD8</td>
<td>INT_TO_REAL</td>
</tr>
</tbody>
</table>

### Counters

<table>
<thead>
<tr>
<th>Counter</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNCTR</td>
<td>UPCTR</td>
</tr>
</tbody>
</table>

### Data Move

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARRAY_SIZE</td>
<td>BUS_RMW_WORD</td>
</tr>
<tr>
<td>ARRAY_SIZE_DIM1</td>
<td>BUS_TS_BYTE</td>
</tr>
<tr>
<td>ARRAY_SIZE_DIM2</td>
<td>BUS_TS_WORD</td>
</tr>
<tr>
<td>BLK_CLR_WORD</td>
<td>BUS_WRT_BYTE</td>
</tr>
<tr>
<td>BLKMOV_DINT</td>
<td>BUS_WRT_DWORD</td>
</tr>
<tr>
<td>BLKMOV_DWORD</td>
<td>BUS_WRT_WORD</td>
</tr>
<tr>
<td>BLKMOV_INT</td>
<td>COMM_REQ</td>
</tr>
<tr>
<td>BLKMOV_REAL</td>
<td>DATA_INIT_ASCII</td>
</tr>
<tr>
<td>BLKMOV_UINT</td>
<td>DATA_INIT_COMM</td>
</tr>
<tr>
<td>BLKMOV_WORD</td>
<td>DATA_INIT_DINT</td>
</tr>
<tr>
<td>BUS_RD_BYTE</td>
<td>DATA_INIT_DLAN</td>
</tr>
<tr>
<td>BUS_RD_DWORD</td>
<td>DATA_INIT_DWORD</td>
</tr>
<tr>
<td>BUS_RD_WORD</td>
<td>DATA_INIT_INT</td>
</tr>
<tr>
<td>BUS_RMW_BYTE</td>
<td>DATA_INIT_LREAL</td>
</tr>
<tr>
<td>BUS_RMW_DWORD</td>
<td>DATA_INIT_UINT</td>
</tr>
</tbody>
</table>

### Logical Programs and Blocks

**LD Editor**
### Data Table

<table>
<thead>
<tr>
<th>Logic Programs and Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LD Editor</strong></td>
</tr>
</tbody>
</table>

#### Data Table

<table>
<thead>
<tr>
<th>Function</th>
<th>Type</th>
<th>Instruction Type</th>
<th>Condition</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARRAY_MOVE_BOOL</td>
<td>FIFO_WRT_WORD</td>
<td>SEARCH_GE_UINT</td>
<td>SEARCH_NE_DINT</td>
<td></td>
</tr>
<tr>
<td>ARRAY_MOVE_BYTE</td>
<td>LIFO_RD_DINT</td>
<td>SEARCH_GE_WORD</td>
<td>SEARCH_NE_DWORD</td>
<td></td>
</tr>
<tr>
<td>ARRAY_MOVE_DINT</td>
<td>LIFO_RD_DWORD</td>
<td>SEARCH_GT_BYTE</td>
<td>SEARCH_NE_INT</td>
<td></td>
</tr>
<tr>
<td>ARRAY_MOVE_JUMP</td>
<td>LIFO_RD_INT</td>
<td>SEARCH_GT_DINT</td>
<td>SEARCH_NE_UINT</td>
<td></td>
</tr>
<tr>
<td>ARRAY_MOVE_UINT</td>
<td>LIFO_RD_UINT</td>
<td>SEARCH_GT_DWORD</td>
<td>SEARCH_NEWORD</td>
<td></td>
</tr>
<tr>
<td>ARRAY_MOVE_WORD</td>
<td>LIFO_WRT_DINT</td>
<td>SEARCH_GT_UINT</td>
<td>SEARCH_NE_WORD</td>
<td></td>
</tr>
<tr>
<td>ARRAY_RANGE_JUMP</td>
<td>LIFO_WRT_DINT</td>
<td>SEARCH_GT_DWORD</td>
<td>SORT_INT</td>
<td></td>
</tr>
<tr>
<td>ARRAY_RANGE_UINT</td>
<td>LIFO_WRT_DINT</td>
<td>SEARCH_GT_DWORD</td>
<td>TBL_READ_DINT</td>
<td></td>
</tr>
<tr>
<td>FIFO_RD_DINT</td>
<td>SEARCH_EQ_DINT</td>
<td>SEARCH_EQ_DINT</td>
<td>TBL_READ_DWORD</td>
<td></td>
</tr>
<tr>
<td>FIFO_RD_DWORD</td>
<td>SEARCH_EQ_DINT</td>
<td>SEARCH_EQ_DINT</td>
<td>TBL_READ_DWORD</td>
<td></td>
</tr>
<tr>
<td>FIFO_RD_INT</td>
<td>SEARCH_EQ_INT</td>
<td>SEARCH_EQ_INT</td>
<td>TBL_READ_DWORD</td>
<td></td>
</tr>
<tr>
<td>FIFO_RD_UINT</td>
<td>SEARCH_EQ_UINT</td>
<td>SEARCH_EQ_UINT</td>
<td>TBL_READ_DWORD</td>
<td></td>
</tr>
<tr>
<td>FIFO_RD_WORD</td>
<td>SEARCH_EQ_WORD</td>
<td>SEARCH_EQ_WORD</td>
<td>TBL_READ_DWORD</td>
<td></td>
</tr>
<tr>
<td>FIFO_WRT_DINT</td>
<td>SEARCH_EQ_DINT</td>
<td>SEARCH_EQ_BYTE</td>
<td>TBL_WRITE_DINT</td>
<td></td>
</tr>
<tr>
<td>FIFO_WRT_DWORD</td>
<td>SEARCH_EQ_DINT</td>
<td>SEARCH_EQ_DINT</td>
<td>TBL_WRITE_DWORD</td>
<td></td>
</tr>
<tr>
<td>FIFO_WRT_DWORD</td>
<td>SEARCH_EQ_DINT</td>
<td>SEARCH_EQ_DINT</td>
<td>TBL_WRITE_DWORD</td>
<td></td>
</tr>
<tr>
<td>FIFO_WRT_DWORD</td>
<td>SEARCH_EQ_DINT</td>
<td>SEARCH_EQ_DINT</td>
<td>TBL_WRITE_DWORD</td>
<td></td>
</tr>
<tr>
<td>FIFO_WRT_DINT</td>
<td>SEARCH_EQ_DINT</td>
<td>SEARCH_EQ_DINT</td>
<td>TBL_WRITE_DWORD</td>
<td></td>
</tr>
<tr>
<td>FIFO_WRT_INT</td>
<td>SEARCH_EQ_DINT</td>
<td>SEARCH_EQ_DINT</td>
<td>TBL_WRITE_DWORD</td>
<td></td>
</tr>
<tr>
<td>FIFO_WRT_UINT</td>
<td>SEARCH_EQ_DINT</td>
<td>SEARCH_EQ_DINT</td>
<td>TBL_WRITE_DWORD</td>
<td></td>
</tr>
<tr>
<td>FIFO_WRT_WORD</td>
<td>SEARCH_EQ_DINT</td>
<td>SEARCH_EQ_DINT</td>
<td>TBL_WRITE_DWORD</td>
<td></td>
</tr>
</tbody>
</table>

#### Math

<table>
<thead>
<tr>
<th>Function</th>
<th>Type</th>
<th>Instruction Type</th>
<th>Condition</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS_DINT</td>
<td>ADD_REAL</td>
<td>DIV_UINT</td>
<td>MUL_MIXED</td>
<td>SUB_DINT</td>
</tr>
<tr>
<td>ABS_INT</td>
<td>ADD_UINT</td>
<td>MOD_DINT</td>
<td>MUL_REAL</td>
<td>SUB_INT</td>
</tr>
<tr>
<td>ABS_LREAL</td>
<td>DIV_DINT</td>
<td>MOD_INT</td>
<td>MUL_UINT</td>
<td>SUB_LREAL</td>
</tr>
<tr>
<td>ABS_REAL</td>
<td>DIV_INT</td>
<td>MOD_UINT</td>
<td>SCALE_DINT</td>
<td>SUB_REAL</td>
</tr>
<tr>
<td>ADD_DINT</td>
<td>DIV_LREAL</td>
<td>MUL_DINT</td>
<td>SCALE_INT</td>
<td>SUB_UINT</td>
</tr>
<tr>
<td>ADD_INT</td>
<td>DIV_MIXED</td>
<td>MUL_INT</td>
<td>SCALE_UINT</td>
<td>SUB_UINT</td>
</tr>
<tr>
<td>ADD_LREAL</td>
<td>DIV_REAL</td>
<td>MUL_LREAL</td>
<td>SCALE_WORD</td>
<td>SUB_LREAL</td>
</tr>
</tbody>
</table>
Program Flow

ARG_PRES_p END H_WIRE LABEL MCRN
CALL ENDMCR JUMP LABELN V_WIRE
COMMENT ENDMCRN JUMPN MCR

Relational

CMP_DINT^ EQ_INT GE_REAL LE_DINT LT_LREAL^ P NE_UINT^
CMP_INT^ EQ_LREAL^ P GE_UINT^ LE_INT LT_REAL RANGE_DINT
CMP_LREAL^ P EQ_REAL GT_DINT LE_LREAL^ P LT_UINT^ RANGE_DWORD^
CMP_REAL^ EQ_UINT^ LE_REAL NE_DINT RANGE_INT
CMP_UINT^ GT_DINT LE_LREAL^ P LT_UINT^ NE_INT RANGE_UINT^
EQ_DATA^ P GE_INT GT_REAL LT_DINT NE_LREAL^ P RANGE_WORD
EQ_DINT GE_LREAL^ P GT_UINT^ LE_INT NE_REAL

Timer Function Blocks

OFDT_HUNDS ONDTR_HUNDS TMR_HUNDS TOP^ P
OFDT_SEC^ ONDTR_SEC^ TMR_SEC^ TON^ P
OFDT_TENTHS ONDTR_TENTHS TMR_TENTHS TP^ P
OFDT_THOUS ONDTR_THOUS TMR_THOUS

VersaMax Micro Motion

BLENDING FIND_HOME GO_HOME JOGGING STOP_MOTION
FBD Editor

The Function Block Diagram (FBD) editor is used to create logic with the Function Block Diagram programming language. FBD is a process language. FBD logic graphically represents the programmed actions performed by a Controller as it executes.

The FBD editor is free form; that is, instructions and text boxes can be placed anywhere on the FBD editor where there is empty space. Sequences of instructions can be wired together horizontally and vertically.

You can work with the FBD editor while offline to edit a disk copy of a project, or you can edit an FBD block of logic online, but this causes the logic to become not equal until you download the FBD block.

You can customize the appearance and behavior of the FBD editor by setting options (see page 26).

An FBD block is a named section of FBD Logic that is compiled and downloaded to the Controller represented by the associated target.

To customize the FBD editor

1. In the Options tab of the Navigator, expand the Editors folder, then expand the Function Block Diagram folder.

2. Right-click the Colors and Preferences page and choose Properties.

   The configurable settings appear as properties in the Inspector.

3. In the Inspector, adjust settings as required.

To create an FBD block

1. In the Project tab of the Navigator, right-click the Program Blocks folder, point to New, and then choose FBD Block.

   A new FBD block with a default name is created.

2. Rename the block as desired.
To open an FBD block for editing

- In the Project tab of the Navigator, double-click an FBD block.
  The block opens in the FBD editor.

**Note:** You can have multiple blocks open for editing. To navigate to another open FBD block, click the tab that displays its name at the top of the editor window.

**Working with the FBD Editor Offline**

While in offline mode, there is no live communication between the FBD editor and the target. Most logic development is done while offline. The following diagram illustrates some of the more common operations you can perform using the FBD editor offline.
Insert instructions and Text boxes from the Toolchest or right-click in the FBD

Drag variables from the Navigator into the FBD editor

Use the Inspector to edit the number of inputs for some FBD instructions

The Companion automatically provides information on what you click

Click a connection point of an instruction to draw a wire from one instruction to another

Drag FBD instructions from the Toolchest into an empty space in the FBD editor

Hover over a variable to see information related to the variable

Displays a list of the places where the selected variable is used or a list of the blocks the selected block is called from

Right-click in an empty space in the FBD editor and choose an instruction to insert
To insert an instruction

1. In the FBD editor, right-click an empty cell and choose **Insert Instruction**.
   A smart list appears listing all available instruction mnemonics.

2. Choose an instruction mnemonic from the list and press ENTER.

To assign a parameter beside an instruction

There are two methods to assign a parameter to an instruction.

**First method (from the FBD editor)**

1. In the FBD editor, hover the mouse pointer immediately before an input connection point or immediately beyond an output connection point of an FBD instruction, so that the mouse pointer appears as .

2. Double-click the connection point.

3. In the smart list that appears, enter or choose an existing variable, constant, or expression to assign to the instruction parameter.

   In the FBD editor, the instruction may display as follows:

**Note:** Inside the rectangle of the ADD instruction, the "1" indicates the solve order of the instruction. Under the variable `myCounter`, the "1" is the initial value of `myCounter`.

**Second method (outside the FBD editor)**
1. In the Variables tab of the Navigator, or the Data Watch tool, select a variable.

2. Drag the selected variable to the connection point of an FBD instruction.

3. When the mouse pointer appears as , release the left mouse button to assign the variable to a parameter of the FBD instruction.

For example, the following diagram shows that you can assign a parameter to the FBD SUBTRACT instruction at the output connection point named Q.

After you have assigned the parameter, the instruction may appear as follows:

Note: In the above diagram, “61” is the initial value of myResult. “4” inside the rectangle of the FBD instruction indicates its solve order.

Want to know more? From the Help menu, select Search. Look up “solve order”. Select “FBD Editor: an Overview”.
To assign a parameter above an FBD instruction or function block instance

The FBD Call instruction and every function block instance (an instance of a counter, PID, timer, HART utility, or user-defined function block) require a parameter to be assigned immediately above the instruction or instance.

1. In the FBD editor, hover with the mouse pointer immediately above an FBD Call or function block instance so that the mouse pointer appears as .

2. When the mouse pointer appears as , click the left mouse button.
   The instruction or function block instance appears in the FBD editor with the parameter selected, as shown below.

3. Without moving the mouse pointer, click again.
   A smart list appears. If this is an FBD Call instruction, the smart list contains a list of all callable blocks of logic in your target. If this is an FBD function block instance, the smart list contains a list of variables that you can choose from to assign instance data to the function block instance. Normally you would choose or create instance data that is not used by any other function block instance, but you can assign the same instance data to multiple function block instances if you want.

4. In the smart list, enter or choose an existing block of logic, or enter a new variable, or enter or choose an existing variable.
   The block appears as the destination of the FBD Call or the variable is assigned as instance data to the function block instance.

To check (validate) a single FBD block

1. In the Project tab of the Navigator, if expand the target that contains the block to check, and then expand the Logic folder.
2. Expand the Program Blocks folder and then, if the block resides in a user-defined folder, expand the user-defined folder.
3. Right-click the block and choose Check Block.
   Machine Edition* checks the block for errors. Any messages or errors are displayed in the Build tab of the Feedback Zone.
To change the number of inputs for FBD instructions (ADD, AND, MUL, OR, SUB, XOR)

1. In the FBD editor, select an ADD, AND, MUL, OR, SUB, or XOR instruction.
2. In the Inspector, select the Number of Inputs property, and then enter or choose from the list the number of inputs you need.
3. If required, draw a wire or assign a variable or constant to the input and output parameters.

Want to know more? In the online help index, look up “FBD Block”. Click the Related Topics button and choose “FBD Controller logic: an Overview”.

To draw a wire

Notes

- At least two instructions must exist in the FBD editor.
- You cannot draw a wire to or from a Text box.

1. In the FBD editor, hover the mouse pointer above a connection point of an FBD instruction or function block instance.
When the mouse pointer appears as 🔄, you can start a wire. See the diagram below.

2. Hold down the left mouse button, and then drag the mouse pointer to a connection point of another FBD instruction

3. Continue to hold down the left mouse button and drag the mouse pointer to a connection point of another FBD instruction, until the mouse pointer appears as 🔄. This indicates that the current location is a valid end point for the wire. See the diagram below.

4. Release the left mouse button.
The newly drawn wire appears as a solid (analog) or a dashed (discrete) line in the FBD editor. See the diagram below.
To negate an FBD parameter
(For discrete parameters only.)
1. In the FBD editor, assign a BOOL parameter to a connection point of an FBD instruction that requires a BOOL variable.
   The diagram may appear as shown below.

2. Right-click a BOOL variable assigned to a connection point, for example, in the diagram above, BOOL1.
3. In the list box that appears, choose Negate. The diagram now appears as shown below.

   Note: Negating a BOOL parameter while online causes logic to be not equal; turning a BOOL parameter ON/OFF while online does not affect logic equality.

To negate an FBD wire
(For discrete FBD wires only.)
1. If required, draw the FBD wire from a connection point that requires a BOOL variable, to a connection point that also requires a BOOL variable.
If the wire is discrete, it appears as a dotted line ……….

2. Right-click the wire, and then choose **Negate**.
The wire now appears as 0……………….

**To move or duplicate FBD logic**

1. In the FBD editor, select an FBD instruction or Text box.
2. To move it:
   - Click the selection and drag it to a new valid location. When you move an instruction, the parameters and wires assigned to it move along with it.
   - or -
   To duplicate it:
     Press **CTRL** while clicking the selection and drag the selection to where you want to place the duplicate. When you release the mouse button, a duplicate copy of the selected instruction is placed in the new area. When you duplicate an instruction, its parameters are also duplicated; wires are not duplicated.

---

**Want to know more?** From the Help menu, select Search. Paste "editing FBD blocks" (including the quotation marks) and then choose "Editing FBD blocks".

---

**To zoom in or zoom out an FBD block**

**Note:** To zoom in an FBD block display in the FBD editor makes the FBD appear larger; to zoom out an FBD block display makes the FBD appear smaller.

1. Click anywhere in the FBD editor.
2. Do one of the following:
   - In the Inspector, expand the Diagram Settings group of properties, and then set the zoom property.
   - or -
   - To zoom out, press the "-" key. To zoom in, press the "+" key.
**Working with the FBD Editor Online**

When you are online in monitor mode, you can monitor the Controller but cannot change anything in the Controller or edit your logic.

When online in programmer mode, you can edit your logic, make changes in the Controller, and monitor the Controller. PACSystems targets also support Run Mode Store; that is, you can download logic to a running PACSystems.

See “Offline, Online: monitor mode, programmer mode”, on page 44.

**To turn on/off or force a variable**

- In the FBD editor, right-click a BOOL variable anywhere in FBD logic and choose Force ON, Force OFF, Toggle Force, Turn ON, Turn OFF, or Toggle IO.
Logic Programs and Blocks

FBD Editor

Note: Forcing a variable ON or OFF overrides any actions the logic may take during runtime. That is, if a variable is forced OFF (0), but FBD logic is trying to turn it ON (1), it stays OFF.

Want to know more? From the Help menu, select Search. Enter “Working with Variables and FBD Logic Online” (including the quotation marks).

FBD Instructions, Functions, and Function Blocks

(PACSystems* only.) The following is a list of all FBD instructions, functions, and function blocks available in Logic Developer - PLC*.

Want to know more? In the online help index, look up “FBD Instructions”.

Advanced Math

ABS ASIN COS EXPT LOG SQRT
ACOS ATAN EXP LN SIN TAN

Bit Operations

AND OR ROR SHIFTR
NOT ROL SHIFTL XOR

Comparison

CMP GE LE NE
EQ GT LT RANGE

Communication

PNIO_DEV_COMM

Control

DO_IO MASK_IO_INTR PID_ISA SCAN_SET_IO SUSP_IO_INTR
F_TRIG PID_IND R_TRIG SUS_IO SVC_REQ

Counters

DNCTR UPCTR
### Data Move

<table>
<thead>
<tr>
<th>ARRAY_SIZE</th>
<th>BUS_RMW_BYTE</th>
<th>BUS_TS_WORD</th>
<th>MOV</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARRAY_SIZE_DIM1</td>
<td>BUS_RMW_DWORD</td>
<td>BUS_WRT</td>
<td>MOVE_DATA_EX</td>
</tr>
<tr>
<td>ARRAY_SIZE_DIM2</td>
<td>BUS_RMW_WORD</td>
<td>COMM_REQ</td>
<td>MOVE_TO_FLAT</td>
</tr>
<tr>
<td>BUS_RD</td>
<td>BUS_TS_BYTE</td>
<td>FANOUT</td>
<td>SIZE_OF</td>
</tr>
</tbody>
</table>

### Math

<table>
<thead>
<tr>
<th>ADD</th>
<th>MOD</th>
<th>NEG</th>
<th>SUB</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIV</td>
<td>MUL</td>
<td>SCALE</td>
<td></td>
</tr>
</tbody>
</table>

### Program Flow

<table>
<thead>
<tr>
<th>ARG_PRES</th>
<th>CALL</th>
</tr>
</thead>
</table>

### Timer Function Blocks

<table>
<thead>
<tr>
<th>OFDT_HUNDS</th>
<th>ONDTR_HUNDS</th>
<th>TMR_HUNDS</th>
<th>TOF</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFDT_SEC</td>
<td>ONDTR_SEC</td>
<td>TMR_SEC</td>
<td>TON</td>
</tr>
<tr>
<td>OFDT_TENTHS</td>
<td>ONDTR_TENTHS</td>
<td>TMR_TENTHS</td>
<td>TP</td>
</tr>
<tr>
<td>OFDT_THOUS</td>
<td>ONDTR_THOUS</td>
<td>TMR_THOUS</td>
<td></td>
</tr>
</tbody>
</table>

### Type Conversion

<table>
<thead>
<tr>
<th>BCD4_TO_INT</th>
<th>DINT_TO_DWORD</th>
<th>INT_TO_DINT</th>
<th>REAL_TO_DINT</th>
<th>UINT_TO_DINT</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCD4_TO_REAL</td>
<td>DINT_TO_INT</td>
<td>INT_TO_REAL</td>
<td>REAL_TO_INT</td>
<td>UINT_TO_INT</td>
</tr>
<tr>
<td>BCD4_TO_UINT</td>
<td>DINT_TO_LREAL</td>
<td>INT_TO_UINT</td>
<td>REAL_TO_LREAL</td>
<td>UINT_TO_REAL</td>
</tr>
<tr>
<td>BCD8_TO_DINT</td>
<td>DINT_TO_REAL</td>
<td>INT_TO_WORD</td>
<td>REAL_TO_UINT</td>
<td>UINT_TO_WORD</td>
</tr>
<tr>
<td>BCD8_TO_REAL</td>
<td>DINT_TO_UINT</td>
<td>LREAL_TO_DINT</td>
<td>TRUNC_DINT</td>
<td>WORD_TO_INT</td>
</tr>
<tr>
<td>DEG_TO_RAD</td>
<td>DWORD_TO_DINT</td>
<td>LREAL_TO_REAL</td>
<td>TRUNC_INT</td>
<td>WORD_TO_UINT</td>
</tr>
<tr>
<td>DINT_TO_BCD8</td>
<td>INT_TO_BCD4</td>
<td>RAD_TO_DEG</td>
<td>UINT_TO_BCD4</td>
<td></td>
</tr>
</tbody>
</table>
IL Editor

Instruction List (IL) is a programming language specified by the IEC 61131-3 standard. This text language is accumulator-based and much like the assembly languages used for programming microprocessors. The instructions executed by an IL block modify or use an accumulator that is located in Controller memory. Two types of accumulators are defined: one analog accumulator for numeric and bitwise operations and eight boolean accumulators for discrete logic to support eight levels of nested boolean expressions. The IL editor is free-form with an option to apply a standard formatting rule. The appearance and behavior of the IL editor is user-configurable.

Note: Only Series 90*-30, VersaMax* Controllers, and VersaMax* Nano/Micro support IL logic.

To configure accumulators
1. In the Project tab of the Navigator, right-click the Program Blocks folder and choose Properties.
   The Inspector displays the Accumulator Address properties.
2. In the Boolean Start property, enter the reference address of the first of eight Controller memory locations to use for boolean accumulators.
   The ending address is calculated automatically. The memory area must be %T, %M, or %Q.
3. In the Analog Start property, enter the Controller memory locations to use for the analog accumulator.
   The ending address is calculated automatically. The memory area must be %R, %AI or %AQ.

To create an IL block
1. In the Project tab of the Navigator, right-click the Program Blocks folder, point to New, and choose IL Block.
   An empty IL block with the default name "ILBkn" is added to the folder, where n represents a unique number.
2. Rename the block as desired.
To open an IL block for editing

- In the Project tab of the Navigator, right-click an IL Block and choose Open.

The block opens in the IL editor.

**Note:** You can have multiple blocks open for editing. To navigate to another open IL block, click the tab displaying its name at the top of the editor window.

**Working with the IL Editor Offline**

Most project development is carried out while offline from the target Controller. Editing while offline provides maximum flexibility and enables you to interact with the Machine Edition tools as shown in the following diagram.

To insert an instruction

1. In the IL editor, right-click and choose **Insert Keyword**.

A smart list appears listing all available instruction mnemonics.

2. From the list, select an instruction and press **ENTER**.

The instruction is inserted in your logic.
To assign operands to an instruction
1. In the IL editor, right-click and choose Insert Variable.
   A smart list appears showing all your defined variables.
2. Type, or choose from the list, a variable name or reference address and then press ENTER.
   The name appears in your logic.
   **Note:** If you entered a reference address or a new variable name, you must create a variable from it.

To create a variable from a reference address
- In the IL editor, right-click a reference address, point to Create “name” as, and then choose a data type.
  A variable is created and a default name is applied. For example, if the reference address is %R0032, the auto-created variable is named R00032.

To create a variable from a name
1. In the IL editor, right-click a name, point to Create “name” as, and then choose a data type.
   A variable is created with the name you right-clicked.
2. Map the variable to Controller memory. (See page 24.)

To move or duplicate IL logic
1. In the IL editor, select a range of logic.
2. To move it, click the selection and drag it to a new location. To duplicate it, press CTRL while clicking the selection, and drag the selection to where you want to place the duplicate.
   When you release the mouse button, the selection is respectively moved to the new location or a duplicate copy of the selected logic is placed in the new area.

To insert an inline comment
1. In the IL editor, click where you want to insert an inline comment.
2. Type an apostrophe (’) followed by comment text.
3. Press ENTER to complete the comment.

To insert a block comment
1. In the IL editor, click where you want to insert a block comment.
2. Type (* followed by comment text.
Logic Programs and Blocks

IL Editor

A block comment can contain any number of characters and can span multiple lines.

3. Type ")" to complete the block comment.

To reformat IL logic

- In the IL editor, right-click and choose **Beautify Source**.

The entire content of the IL editor is reformatted according to the default formatting (indentation) rules.

**Working with the IL Editor Online**

When online in monitor mode, you can monitor the Controller but cannot change anything in the Controller or edit your logic.

When you are online in programmer mode, you can edit your logic, make changes in the Controller, and monitor the Controller. Many GE targets support Run Mode Store; that is, you can download logic to them when they are running.

See “Offline, Online: monitor mode, programmer mode”, on page 44.
To monitor a data value

- Click anywhere in the IL editor and hover the mouse pointer over a variable. A tooltip appears, showing the variable’s current value. This value, however, does not update automatically: you must move the mouse pointer away and back over the variable to update the value.

To change a BOOL variable’s state

- In the IL editor, right-click the BOOL variable whose value you want to change and choose Turn On, Turn Off, or Toggle IO.
  The state of the variable in the target Controller changes when the command is received. It remains in that state until acted on by the Controller’s logic.

To force a BOOL variable’s state

- In the IL editor, right-click a BOOL variable and choose Force On, Force Off, or Toggle Force.
  The state of a forced variable remains unchanged, regardless of any actions by Controller logic.

To remove the force from a BOOL variable

- In the IL editor, right-click a BOOL variable and choose Remove Forces.
  The state of the forced variable is controlled by Controller logic from now on.

**Downloading Online Changes to a GE IP Controller**

You can change your IL logic while online, but you must download the changes to the Controller to maintain equality. Some GE Controllers support the downloading of changed IL logic even when they are executing.

To make changes to IL logic and download them to a running target Controller (if the target Controller supports it)

1. While online to a target Controller, make a change to IL logic.
2. In the Project tab of the Navigator, right-click the target and choose Download to Controller.
  You are prompted to confirm a Run Mode Store. That is, the altered IL block will be downloaded to the running Controller without stopping it.

---

Want to know more? In the online help index, look up “Downloading to running GE Controllers”.
IL Instructions

The following is a list of all IL instructions available in Logic Developer - PLC*. Companion help indicates which Controllers support the instructions.

**Basic Instructions**

<table>
<thead>
<tr>
<th>ADD</th>
<th>GT</th>
<th>LT</th>
<th>OR</th>
<th>ST_DINT</th>
<th>XOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>AND</td>
<td>LD_BOOL</td>
<td>MOD</td>
<td>ORN</td>
<td>ST_INT</td>
<td>XORN</td>
</tr>
<tr>
<td>ANDN</td>
<td>LD_ENO</td>
<td>MUL</td>
<td>PT</td>
<td>ST_REAL</td>
<td></td>
</tr>
<tr>
<td>DIV</td>
<td>LD_INT</td>
<td>NE</td>
<td>R</td>
<td>ST_WORLD</td>
<td></td>
</tr>
<tr>
<td>EQ</td>
<td>LDN_BOOL</td>
<td>NOT</td>
<td>S</td>
<td>STN_BOOL</td>
<td></td>
</tr>
<tr>
<td>GE</td>
<td>LE</td>
<td>NT</td>
<td>ST_BOOL</td>
<td>SUB</td>
<td></td>
</tr>
</tbody>
</table>

**Advanced Math**

| ACOS | COS | LN | SORT_DINT | TAN |
| ASIN | EXP | LOG | SORT_INT |   |
| ATAN | EXPT | SIN | SORT_REAL |   |

**Bit Operations**

| AND_WORD | BIT_SEQ | MASK_COMP_DWORD | OR_WORD | SHIFTL_WORD |
| BIT_CLR_WORD | BIT_SET_WORD | MASK_COMP_WORD | ROL_WORD | SHIFTR_WORD |
| BIT_POS_WORD | BIT_TEST_WORD | NOT_WORD | ROR_WORD | XOR_WORD |

**Communication**

MODBUS_TCP_RW (VersaMax Micro CPUs with firmware version 4.00 or later)

**Control**

| DO_IO | DRUM | PID_IND | PID_IsA | SER | SVC_REQ |

**Conversions**

| BCD4_TO_INT | DINT_TO_REAL | RAD_TO_DEG | REAL_TO_WORD | WORD_TO_REAL |
| BCD4_TO_REAL | INT_TO_BCD4 | REAL_TO_DINT | TRUNC_DINT |   |
| DEG_TO_RAD | INT_TO_REAL | REAL_TO_INT | TRUNC_INT |   |

**Counters**

| DNCTR | UPCTR |

A superscript V (*) indicates instructions supported only for VersaMax Controllers.
Data Move

- BLK_CLR_WORD
- BLKMOV_REAL
- COMM_REQ
- MOVE_BOOL
- MOVE_REAL
- SHFR_BIT
- BLKMOV_INT
- BLKMOV_WORD
- MOVE_INT
- MOVE_WORD
- SHFR_WORD

Data Table

- ARRAY_MOVE_BOOL
- SEARCH_EQ_INT
- SEARCH_GT_DINT
- SEARCH_LT_BYTE
- SEARCH_NE_WORD
- ARRAY_MOVE_BYTE
- SEARCH_EQ_WORD
- SEARCH_GT_INT
- SEARCH_LT
- ARRAY_MOVE_DINT
- SEARCH_EQ_BYTE
- SEARCH_GT_WORD
- SEARCH_LT_DINT
- ARRAY_MOVE_WORD
- SEARCH_EQ_DINT
- SEARCH_GT_BYTE
- SEARCH_LT_INT

Math

- ADD_DINT
- DIV_DINT
- MOD_DINT
- MUL_DINT
- SCALE_WORD
- SUB_REAL
- ADD_INT
- DIV_INT
- MOD_INT
- MUL_REAL
- SCALE_INT
- SUB_INT
- ADD_REAL
- DIV_REAL
- MUL_REAL
- SCALE_WOR
- SUB W

Program Flow

- CAL
- CALC
- END_MCRN
- JMPC
- MCRN
- RETC
- CALC
- END
- JMP
- JMPCN
- RET
- RETCN

Relational

- EQ_DINT
- GE_DINT
- GT_DINT
- LE_DINT
- LT_DINT
- NE_DINT
- RANGE_DINT
- EQ_INT
- GE_INT
- GT_INT
- LE_INT
- LT_INT
- NE_INT
- RANGE_INT
- EQ_REAL
- GE_REAL
- GT_REAL
- LE_REAL
- LT_REAL
- NE_REAL
- RANGE_WOR

Timer Function Blocks

- OFDT_HUNDS
- ONDTR_HUNDS
- TMR_HUNDS
- OFDT_TENTHS
- ONDTR_TENTHS
- TMR_TENTHS
- OFDT_THOUS
- ONDTR_THOUS
- TMR_THOUS
ST EDITOR

Structured Text logic is a programming language specified by the IEC 61131-3 standard. The Structured Text (ST) Editor is a free-form text editor for editing Structured Text logic in the Machine Edition environment. With the ST editor, you can work on a disk copy of a structured text block (offline) or monitor the execution of an ST block running in the Controller (online). You can edit an ST block online, but this causes the logic to become not equal until you download the ST block.

ST is a high-level language that uses various operators and functions. ST logic is supported by PACSystems Controllers.

To customize the ST Editor

1. In the Options tab of the Navigator, expand the Editors folder and then the Text folder.
2. Right-click a page (Colors or Preferences), and choose Properties. The configurable settings appear as properties in the Inspector.
3. In the Inspector, adjust the settings as required.

To create an ST block

1. In the Project tab of the Navigator, expand the Logic folder, right-click the Program Blocks folder or a user-defined folder, point to New, and then choose ST block. A new ST block with a default name is created under the Program Blocks folder or under a user-defined folder.
2. (Optional.) Rename the block.

To create a parameterized ST block

1. Create an ST block.
2. Do one of the following:
   - In the Inspector, set parameters for the block.
   - or -
   - Set the block’s Block Type property to Parameterized Block, and then optionally schedule the parameterized block.

Notes:
Logic Programs and Blocks

ST Editor

- By assigning parameters to the block, its Block Type property is automatically changed to Parameterized Block.
- A parameterized block cannot have both interrupt scheduling and parameters.

Want to know more? In the online help index, look up “Parameterized Block”.

To open an ST block for editing

- In the Project tab of the Navigator, under the Logic folder, under the Program Blocks folder, or under a user-defined folder, double-click an ST block.

The ST block opens in the ST editor.

Working with the ST Editor Offline

The ST editor interacts with the Machine Edition tools to provide maximum flexibility when editing a program. The following diagram illustrates some of the operations you can perform.
To insert an ST variable or keyword

1. In the ST editor, right-click and choose **Insert Variable** or **Insert Keyword**. A smart list appears, prompting you to enter the name of an existing variable or keyword.
2. Type, or choose from the list, the item you want, and then press ENTER. The variable or keyword appears in the ST editor.

**Notes:**
- Parameters and constants must be manually entered.
- When entering an indirect reference as an operand, prefix the variable name with @ after selecting it from the smart list, for example, @IntVar. Indirect referencing is available for all register references (%R, %P, %L, %W, %AI, and %AQ).

To create a variable from a name

- In the ST editor, right-click a name that was entered as an operand, point to **Create** `[name]` as, and then choose a data type from the list that appears.
A new variable of the specified data type is created and substituted for the name at every occurrence in the ST block. The name of the new variable is identical with the original name.

To insert a line comment

1. In the ST editor, click where you want a line comment to begin.
2. Type an apostrophe (') or `//` followed by comment text.
3. Press ENTER to complete the line comment. The line comment ends when a new line starts.

To insert a block comment

1. In the ST editor, click where you want a block comment to begin.
2. Type `(*) followed by comment text. A block comment can contain any number of characters and can span multiple lines.
3. Type `*) to complete the block comment.
To select a range of ST logic
1. In the ST editor, click at the beginning of the range.
2. Press and hold shift and then click at the end of the range. All logic within the range is selected.
   Tips:
   - You can also click and drag from the beginning of the range to the end.
   - After selecting a range of logic, you can click anywhere in it and drag it to another area in the ST editor.

To move or duplicate ST logic
1. In the ST editor, select a range of logic.
2. To move it, click the selection and drag it to the new location. To duplicate it, press CTRL while clicking the selection, and then drag the selection to where you want to place the duplicate.
   When you release the mouse button, the selection is respectively moved to the new location or a duplicate copy of the selected logic is placed in the new area.

To locate all occurrences of a variable
1. In the Feedback Zone, click the References tab.
2. Click a variable anywhere it appears in ST logic, in the Variables tab of the Navigator, or in the Data Watch tool.
   Each occurrence of the variable in your project (for a given target) is listed in the References tab of the Feedback Zone.
Working with the ST Editor Online

When online in monitor mode, you can monitor the Controller but cannot change anything in the Controller or edit your logic.

When you are online in programmer mode, you can edit your logic, make changes in the Controller, and monitor the Controller. PACSystems* targets also support Run Mode Store; that is, you can download logic to a running PACSystems.

See “Offline, Online: monitor mode, programmer mode”, on page 44.

To view a variable value

- In the ST editor, hover the mouse pointer over a variable to see its value. A tool tip displays the variable name, address, description, data type, and current value. If this is a BOOL variable, the tool tip also displays if it has been forced On (1) or Off (0).
To view the value of an ST parameterized block parameter

1. In the Project tab of the Navigator, expand the Program Blocks folder.
2. If the block that contains the call to the ST parameterized block whose parameter you want to monitor resides in a user-defined folder, expand the user-defined folder.
3. Right-click the block that contains the call and choose Open.
   The block appears in the appropriate editor.
4. In the editor, right-click the block call to the ST parameterized block and choose Open Block.
   The called ST parameterized block appears in the ST editor.
5. In the ST editor, hover the mouse pointer over the parameterized block parameter (variable) to see its value.
   Depending on the context of the parameter, the tool tip displays the following items:
   - If the text refers to a parameter instead of a variable, and the block was opened from a CALL instruction (in an LD, ST, or FBD block), then if possible, the parameter is associated with the variable used in the call the block was opened from. In this case, the tool tip displays the parameter name, address of the variable, description of the parameter, data type, and value of the variable.
   - If the text refers to a parameter instead of a variable, and the block was not opened from a CALL instruction (in an LD, ST, or FBD block), then the tool tip displays the parameter name, description of the parameter, and data type only.
   In either case, if this is a forced BOOL variable, the tool tip also displays if the variable has been forced On (1) or Off (0).
   Note: The tool tip displays a variable's current value only if logic is equal; the tool tip doesn't display the current value if logic is not equal.

To change a BOOL variable's state

- In the ST editor, right-click the BOOL variable whose value you want to change and choose Turn On, Turn Off, or Toggle IO.
  The state of the variable in the target Controller changes when the Controller receives the command.
To force a BOOL variable’s state
- In the ST editor, right-click a BOOL variable and choose Force On, Force Off, or Toggle Force.
  The state of a forced variable remains unchanged, regardless of any actions by Controller logic.

To remove the force from a BOOL variable
- In the ST editor, right-click a BOOL variable and choose Remove Forces.
  The state of the unforced variable is controlled by Controller logic from now on.

ST Statements, Functions, and Function Blocks
The following is a list of all ST statements, functions, and function blocks available in Logic Developer - PLC*. InfoViewer Help indicates which firmware version of PACSystems supports the statements, functions, or function blocks.

Statements

<table>
<thead>
<tr>
<th>:= (Assignment)</th>
<th>Comment</th>
<th>Function block invocation</th>
<th>Repeat ... until</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block call</td>
<td>Exit</td>
<td>Function call</td>
<td>Return</td>
</tr>
<tr>
<td>Case</td>
<td>For ... do</td>
<td>If</td>
<td>While ... do</td>
</tr>
</tbody>
</table>

Advanced Math

<table>
<thead>
<tr>
<th>ACOS</th>
<th>ATAN</th>
<th>EXP</th>
<th>LN</th>
<th>SIN</th>
<th>SQRT_REAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACOS_REAL</td>
<td>ATAN_REAL</td>
<td>EXP_REAL</td>
<td>LN_REAL</td>
<td>SIN_REAL</td>
<td>TAN_LREAL</td>
</tr>
<tr>
<td>ACOS_REAL</td>
<td>ATAN_REAL</td>
<td>EXP_REAL</td>
<td>LN_REAL</td>
<td>SIN_REAL</td>
<td>TAN_LREAL</td>
</tr>
<tr>
<td>ASIN</td>
<td>COS</td>
<td>EXPT, **, or ^</td>
<td>LOG</td>
<td>SQRT_DINT</td>
<td>TAN_REAL</td>
</tr>
<tr>
<td>ASIN_REAL</td>
<td>COS_REAL</td>
<td>EXPT_REAL</td>
<td>LOG_REAL</td>
<td>SQRT_INT</td>
<td>SQRT_LREAL</td>
</tr>
</tbody>
</table>

Bitwise Operators

| AND             | NOT     | OR        | XOR      |

Communication

PNIO_DEV_COMM
Logic Programs and Blocks
ST Editor

Control
DO_IO MASK_IO_INTR SCAN_SET_IO SUSP_IO_INTR SWITCH_POS
F_TRIG R_TRIG SUS_IO SVC_REQ

Conversions
ANGLES LREAL to DINT or REAL
BCD4 to INT, REAL, or UINT REAL to DINT, INT, LREAL, or UINT
BCD8 to DINT or REAL TRUNC_DINT, TRUNC_INT
DINT to BCD8, DWORD, INT, LREAL, REAL, or UINT UINT to BCD4, DINT, INT, REAL, or WORD
DWORD to DINT WORD to INT or UINT
INT to BCD4, DINT, REAL, UINT, or WORD

Data Move
ARRAY_SIZE ARRAY_SIZE_DIM2 MOVE_DATA_EX SIZE_OF
ARRAY_SIZE_DIM1 COMM_REQ MOVE_TO_FLAT

Program Flow
ARG_PRES

Math Functions
ABS_DINT ABS_LREAL SCALE_DINT SCALE_UINT
ABS_INT ABS_REAL SCALE_INT

Math Operators
+ (addition) - (subtraction) • (negation) * (multiplication) / (division) mod (modulo)

Relational Operators
= (Equal) >= (Greater than or equal) > (Greater than) <= (Less than or equal) < (Less than) <> != (Not equal)

Timer Function Blocks
TOF TON TP
C BLOCKS

A C block is an independent section of executable code written in the C programming language that is downloaded to and executed on the target Controller. C blocks are created externally using GE’s C Programming Toolkit and then imported into a project. A C block compiled for PACSystems* has a .gefelf extension. A C block compiled for Series 90*-70 or Series 90*-30 has a .exe extension. C Blocks can be called as a subroutine from another block (LD, ST, FBD, or IL) but cannot call another block.

For more information on developing C Blocks, refer to C Programmer’s Toolkit for Series 90 Controllers (GFK-0646) and PACSystems C Toolkit User’s Guide (GFK-2259).

Working with C blocks

To import a C block

1. In the Project tab of the Navigator, right-click the Program Blocks folder and choose Add C block.

The Open dialog box appears.

2. Browse to the .exe or .gefelf file you want to import and click Open.

The selected file is added to the Program Blocks folder with the same name as the .exe or .gefelf file.

Note: You can then move the C Block to a user-defined folder.

Want to know more? In the online help index, look up “C block” and “User-Defined Folder”.
Logic Programs and Blocks
C Blocks

To set a C block’s parameters

Note: This procedure applies only to C blocks that were written to require parameters for use on PACSystems* or Series 90*-70 targets.

1. In the Project tab of the Navigator, right-click the C block and choose Properties.

   The Inspector displays the block’s properties.

2. In the Inspector, select the Parameters property and click .

   The Parameters dialog box appears.

3. Type a name and description for each required input and output parameter.

   For information on the required parameters, refer to the written documentation for the C block. The names you enter will display in the CALL instruction that calls the C block. The names and descriptions will display in a tooltip when you hover the mouse pointer over the CALL instruction.

Want to know more? In the online help index, enter “Parameterized” and select the “C Block” help topic.
C PROGRAMS

Note: C programs are supported only on Series 90*-70 CPUs, firmware release 6.00 and later.

A C program is an independent section of executable code, written in C language, that is downloaded to and executed on the associated target Controller. To develop a C program, use GE's C Programmer's Toolkit for Series 90 Controllers User's Manual (GFK-0646). You then import it into a project.

A C program has access to all the % reference tables of the Controller except for the _MAIN LD block’s %P memory and the %L memory of any other LD block. A C program can also call any of the numerous Controller-embedded functions that are included in the C Programmer’s Toolkit.

A C program cannot be called as a subroutine. Execution is controlled only through scheduling. A C program cannot call a block as a subroutine.

By setting the parameters of a C program, you enable it to access memory directly. When a C program begins to execute, it reads the data for all the parameters and makes a copy of the data. If the C program’s execution is interrupted or time-sliced over multiple scans, the C program, when it resumes execution, uses the copy of the data that it made when it began to execute.

C programs can coexist with a main program and with State Logic on a Series 90-70 Controller. For such a target, the State Logic is compiled into a C program. This C program is named _MAIN if there is no LD logic; however, if State Logic coexists with LD logic, the State Logic is compiled into a C program named _STATE and one LD block must be named _MAIN.
Working with C Programs

Setting a C program’s parameters

Note: You must provide a list of all the input and output parameters that the C program requires. The main program does not use parameters.

1. In the Project tab of the Navigator, right-click a C program and choose Properties.
   The Inspector displays the C program’s properties.
2. In the Inspector, select the Parameters property and click...
   The Parameters dialog box appears.
3. On the Input and Output tabs, enter up to 8 input and up to 8 output parameters. Each parameter has its own row on the tab. For each parameter, double-click the following cells and enter the required data:
   - **Name:** The parameter’s name.
   - **Type:** The parameter’s data type.
   - **Length:** The length of the input or output reference.
   - **Variable:** The first data item associated with the parameter.
   - **Description:** (Optional.) The parameter’s description.

Want to know more? In the online help index, choose “C Program”.
LD Diagnostic Logic Blocks

(PACSystems® RX3i firmware version 5.60 and later.)

A PACSystems RX3i target contains the Diagnostic Logic Blocks folder, which contains two empty subfolders: the Active Blocks folder and the Inactive Blocks folder.

Active Blocks folder

The Active Blocks Folder can contain the following:

- One or more active LD DLBs.
- Zero through three Cam profiles associated with an LD DLB.
- One or more user-defined folders.

Inactive Blocks folder

The Inactive Blocks Folder can contain the following:

- One or more inactive LD DLBs.
- One or more user-defined folders.

Active LD Diagnostic Logic Blocks

An active LD Diagnostic Logic Block (DLB) is a named section of ladder logic that is compiled and downloaded to the Controller.

Cam profiles associated with an active LD DLB are stored in the Cam profiles folder and are also downloaded to the Controller.

The contents of an active LD DLB are edited with the LD editor.

Immediately before an upload from the Controller is performed, all active LD DLBs and their associated Cam profiles are moved to the Inactive Blocks folder. This causes all active LD DLBs to become inactive LD DLBs prior to the upload, because the variables being used by the active LD DLBs may not exist in your project after the upload completes.

After an upload from the Controller, you can activate an LD DLB. Do this in the Navigator by right-clicking an inactive LD DLB and choosing Activate.
LD Diagnostic Logic Blocks

Every LD DLB, whether active or inactive, has an associated Published Variable Table (PVT) that contains the local published variables of the LD DLB. If an LD DLB has no local variables, then its PVT is empty.

Inactive LD Diagnostic Logic Blocks

An inactive LD Diagnostic Logic Block (DLB) is a named section of ladder logic that is inactive and for storage only. An inactive LD DLB cannot be viewed, edited, compiled, or downloaded to the Controller. Cam profiles associated with an inactive LD DLB also cannot be viewed.

To edit, view, or download the contents of an inactive LD DLB, or to edit or view a Cam Profile associated with an inactive LD DLB, you must first activate the LD DLB. To do this, in the Navigator, right-click an inactive LD DLB and choose Activate.

To create an active LD Diagnostic Logic Block

1. In the Project tab of the Navigator, expand the PACSystems RX3i target where you want to create the active LD Diagnostic Logic Block (DLB), expand the Diagnostic Logic Blocks folder, and then expand the Active Blocks folder.

2. If the active LD DLB is to reside in a user-defined folder, then, if required, add the user-defined folder.

3. Right-click the folder where you want to create the active LD DLB, point to New, and then choose Block.

The active LD DLB is created and its unique default name is highlighted.

Each active LD DLB contains a Cam Profiles folder, where you can add, edit, or import from one through three Cam profiles.

4. (Optional.) Rename the new active LD DLB.

All LD DLBs in the parent folder appear in alphabetical order.

Want to know more? In the online help index, look up “Active LD Diagnostic Logic Block (DLB)”.

Want to know more? In the online help index, look up “Inactive LD Diagnostic Logic Block (DLB)” and “Working with LD Diagnostic Logic Blocks (DLBs)”.

Want to know more? In the online help index, look up “Working with LD Diagnostic Logic Blocks”.
PROFINET Support

(PACSystems* RXi; PACSystems* RX3i with firmware version 7.00 and later.)

Hardware

GE IP PROFINET Controllers (PNCs) support the following PROFINET devices:

<table>
<thead>
<tr>
<th>VersaMax</th>
<th>VersaMax IP</th>
<th>VersaPoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSTi</td>
<td>PAC8000</td>
<td>General Electric AF6</td>
</tr>
<tr>
<td>Third-party</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A PNC is connected by Ethernet to PROFINET devices, each of which has its own modules and possibly submodules.

Software

In Machine Edition, the PROFINET devices connected to a PNC are shown as nodes under the PNC in the Hardware Configuration of a PACSystems RXi or PACSystems RX3i target in a Machine Edition project. The illustration on the next page highlights the target names and the PNCs.

Configuration data

The data required to configure the PROFINET devices connected to the PROFINET Controller is contained in GSDML files provided by the vendors in accordance with PROFINET standards.

Redundancy

PACSystems RX3i Controllers support Hot Standby CPU redundancy with PROFINET I/O. In that setup, two RX3i Controllers are capable of controlling the same set of PROFINET devices. (For the minimum PACSystems RX3i firmware version requirement, refer to online help.)

Want to know more? In the online help index, look up “PROFINET Devices Catalog”, “RXi PROFINET Controller”, “IC695PNC001: RX3i PROFINET Controller (2 SFP)”, “Supported PROFINET devices”, “GSDML files: Overview”, “configuring PROFINET networks”, and “LAN View: Overview (PROFINET)”. 
Ethernet Global Data (EGD)

Ethernet Global Data is a mechanism that enables one CPU, referred to as a producer, to share a portion of its internal memory with one or more other CPUs, referred to as consumers, at a scheduled periodic rate. Such a snapshot of internal memory, mediated by an Ethernet interface, is referred to as an exchange. An exchange is identified by a unique combination of three identifiers:

- The Producer ID (the producer's IP address)
- The Exchange ID (the exchange's identifier)
- The Adapter Name (the Ethernet interface identifier)

Exchanges vs. Pages

Some EGD tools group multiple produced exchanges into a page, and some EGD tools can consume such pages. Logic Developer - PLC*, however, does not consume or produce pages. It consumes and produces exchanges.

- When you add a consumed exchange in Logic Developer - PLC, if an EGD tool has published produced pages that contain multiple exchanges to the EGD Configuration Server, the produced exchanges are extracted from the pages and presented for your selection as exchanges.
- Logic Developer - PLC does not group multiple produced exchanges into a produced page. EGD tools that consume pages see Logic Developer - PLC produced exchanges as pages with only one exchange each.

Integration with the EGD Configuration Server

Logic Developer - PLC* is integrated with the EGD Configuration Server. The EGD Configuration Server is a central repository of EGD configuration information used to facilitate the sharing of information between EGD tools in order to assist with the configuration of EGD.

After configuring a produced exchange on any EGD tool (such as the EGD component in Logic Developer - PLC), you can publish the configuration of the produced exchanges to the EGD Configuration Server. After this, Logic Developer - PLC, when connected to the server, can obtain the produced exchange configuration from the server and automatically create a consumed
exchange that matches the produced exchange exactly. At this point, the only thing left to do is to map the variables of the exchange to memory on the EGD consumer, or to replace some or all of the unmapped variables with mapped variables that already exist on the EGD consumer.

The EGD Configuration Server provides extra EGD validation that is unavailable without the server.

The EGD Configuration Server supports EGD Signatures. When a PACSystems* RX7i or PACSystems* RX3i is configured to use signatures, then every scan, it compares the signature of every consumed exchange with the signature of its corresponding produced exchange to determine whether the configuration of the consumed exchange matches the configuration of the produced exchange; if not, the exchange is not consumed.

The EGD Configuration Server tracks which produced exchanges published on the server are consumed by EGD devices connected to the server. You can obtain an Unconsumed Data report. Other reports compare the produced or consumed exchange information on your computer with that on the server.

The EGD Configuration Server is not the authoritative source of the EGD configuration. It does not store past versions of EGD configurations. It contains only the latest snapshots of the EGD configurations published to it. The EGD tools are the authoritative source, while the server enables the tools to share information. It is possible to delete the server contents completely and restore them again by using the tools in a two-step process: publish the authoritative information from all tools to the server, and have each tool read the information it needs from the server and confirm to the server that it has received it.

Want to know more? In the online help index, look up "Ethernet Global Data in Logic Developer - PLC*: an Overview" and "EGD Configuration Server: Integration with Logic Developer - PLC*".

Integration with the EGD Management Tool

Logic Developer - PLC* is integrated with the EGD Management Tool (EMT). You can open the EMT from within Logic Developer - PLC and vice-versa.

The EMT provides a high-level view of the EGD system published on the EGD Configuration Server, including a graphical display of individual EGD systems. The EMT provides validation of the EGD configuration.

The EMT enables the EGD configuration of devices by launching the appropriate EGD Configuration Tool from the graphical display.

Want to know more? Refer to the online help in the EGD Management Tool.
Logic Developer - PLC* implementation of EGD: the EGD component

The first step in using Ethernet Global Data (EGD) in a GE IP Controller or ENIU target consists in adding the EGD component. In a Controller target, you can do so even if the CPU presently configured in the target does not support EGD, as long as the target belongs to one of the Controller families that support EGD. This flexibility enables you to start configuring EGD even if the hardware specifications of your system have not been determined.

To add the EGD component

1. In the Project tab of the Navigator, right-click a GE IP Controller target that belongs to one of the Controller or ENIU family types that support Ethernet Global Data (EGD), point to Add Component, and choose Ethernet Global Data.

The Ethernet Global Data folder is added to the target. The folder contains two empty folders: Consumed Exchanges and Produced Exchanges.

Note: For Ethernet modules that support EGD uploads, the EGD component is automatically added when you upload the Hardware Configuration and EGD configuration. The EGD component is also automatically added to a target when you import a folder that contains an Ethernet Global Data (EGD) configuration or import a Hardware Configuration that contains an EGD configuration. In such cases, the Consumed Exchanges and Produced Exchanges folders are automatically populated.

To install the EGD Configuration Server

1. Insert the Machine Edition 8.00 install disk into the computer on which you want to install the EGD Configuration Server.

The computer must be connected to an Ethernet network. It may be local or remote.

2. On the Machine Edition 8.00 install disk, navigate to the Install folder.


The EGD Configuration Server is installed on the computer.

Note: The client library required for your copy of Logic Developer - PLC* to communicate with the EGD Configuration Server is automatically installed as part of the installation of Logic Developer - PLC.
To install the EGD Management Tool (EMT) on your computer
1. Insert the Machine Edition* 8.00 install disk into your computer.
2. On the Machine Edition 8.00 install disk, navigate to the Install folder.
   The EGD Management Tool is installed on your computer.

To configure communications with the EGD Configuration Server
1. In the Options tab of the Navigator, expand the Machine Edition folder.
2. Right-click the EGD page and choose Properties.
   The Inspector displays the EGD options.
3. In the Inspector, configure the Host Name option. For help on this or any option, select the option and look up the help that automatically appears in the Companion. To open the Companion, press SHIFT+F11.
4. (Optional.) Configure the Local Server Cache Path and Timeout options.

To configure a Logic Developer - PLC* target to use the EGD Configuration Server
1. In the Project tab of the Navigator, expand the target.
2. Right-click the Ethernet Global Data folder and choose Properties.
   The Inspector displays the folder's properties.
3. Ensure that the Use Configuration Server property is set to True and configure the properties below it.

To add a new produced exchange and configure it
1. In the Project tab of the Navigator, expand the target's Ethernet Global Data folder.
2. Right-click the Produced Exchanges folder and choose New.
   A new produced exchange appears with a default name. The new produced exchange is invalid.
3. (Optional.) Enter a name that is more meaningful than the default name.
4. Right-click the produced exchange and choose Properties.
   The Inspector displays the exchange's properties.
5. In the Inspector, configure the Destination property and other properties. If the new produced exchange is now valid, its icon looks like this: .
6. In the Project tab of the Navigator, right-click the produced exchange and choose Configure.
The EGD Variable editor appears.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Ref Address</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>100001</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

7. In the editor, use the buttons to add, insert, or delete rows.

8. For each row, define the Variable, Ref Address, Length, and/or Description parameters. When you have configured the target to use the EGD Configuration Server, you must specify a Variable for each row; you cannot use memory ranges that have no variables mapped to them. The Ignore parameter is available only for the Status. The Type parameter is read-only.

For help on the editor, click inside the grey space at the top and press F1.

To publish a target’s produced exchanges to the EGD Configuration Server, by using the Validate method

1. In the Project tab of the Navigator, right-click the target and choose Set as active target.
   If the option is unavailable, the target has already been set as the active target.
2. Right-click the target and choose Validate.
   Any errors found in the target’s EGD configuration, Hardware Configuration, or logic are listed in the Build tab of the Feedback Zone.
3. Correct any errors found and repeat step 2 of this procedure.
4. When the Build tab of the Feedback Zone displays no errors, click the Messages tab of the Feedback Zone.
   If the message “EGD Produced Data published” appears, the target’s produced exchanges have been updated (published) from your computer to the EGD Configuration Server.

To publish a target’s produced exchanges to the EGD Configuration Server, by using the Bind and Build method

1. In the Project tab of the Navigator, right-click the target and choose Set as active target.
   If the option is unavailable, the target has already been set as the active target.
2. Expand the target.
3. Right-click the Ethernet Global Data folder and choose **Bind and Build**. Any errors found in the target's EGD configuration are listed in the Messages tab of the Feedback Zone. Some of these errors may pertain to consumed exchanges. If you want context-sensitive help on the errors, we recommend that you use the Validation method instead of the Bind and Build method.

4. If errors are found pertaining to produced exchanges, correct them and repeat step 3 of this procedure. The message “EGD Produced Data published” appears when the target's produced exchanges have been updated (published) from your computer to the EGD Configuration Server.

If an error is found pertaining to a consumed exchange, you typically need to synchronize the consumed exchange.

**To synchronize a consumed exchange on your computer with the corresponding produced exchange published on the EGD Configuration Server:**

1. If any of the following conditions is true, validate the target that contains the producer. (See “Validating a Target”, on page 44.)
   - You have uploaded the Hardware Configuration and EGD configuration from the producer Controller to your computer
   - You have converted the target that contains the producer
   - You have deleted the Ethernet Global Folder from the target that contains the producer and added it again.

   Validating the target that contains the producer updates the EGD Configuration Server with the produced exchange that this consumed exchange consumes.

2. In the Project tab of the Navigator, expand the target.

3. Expand the **Consumed Exchanges** folder.

4. If you have changed the Local Producer ID property of the producer that the consumed exchange consumes from, right-click the consumed exchange and choose **Properties**. Then, in the Inspector, set the Producer ID property on the consumed exchange to the same value as the new Local Producer ID.

5. In the Project tab of the Navigator, right-click the consumed exchange and choose **Synchronize to Server**.

*Want to know more?* In the online help index, look up “Developing EGD systems with the EGD Configuration Server: Possible work flow” and “Troubleshooting EGD.”
The PACMotion Multi-axis Motion controller (PMM) is a high performance, easy-to-use servo motion control module that is closely integrated with the PACSystems RX3i CPU logic solving and communications functions. This versatile motion controller combines the benefits of highly integrated motion and machine logic with the performance, flexibility, and scalability required for advanced machine automation. The open programming environment simplifies motion and machine logic synchronization, and enables real-time performance required for high-speed motion applications.

Expanding the PACMotion node reveals the following PACMotion components used with a PACMotion Digital Motion Control Module (IC695PMM335):

- Cam Profile Library: Contains all of the PACMotion Cam profiles for this target, which cannot be downloaded to a PACMotion module.
- Active Profiles: Contains all of the PACMotion Cam profile aliases for this target, which can be downloaded to a PACMotion module.
- Data Logging Windows: Contains all of the Data Logging Windows for this target. Each window is used to view and print graphs of the values of various IC695PMM335 parameters over time.

PACMotion function blocks and instructions controlling or interacting with a PMM module are supported in LD, FBD, and ST logic. You can drag them into logic from the Toolchest LD Instructions drawer (or FBD Instructions drawer) PACMotion folder.

The following can be used with PACMotion function blocks only:

- Enumerated data types. An enumerated variable stores a value belonging to an enumeration, list, or set of possible values. The value of an enumerated variable can be selected from a drop-down list in the Inspector.
- Reference ID variables (RIVs). A variable of a reference ID data type is used to identify something on which a PACMotion function block instance is used to perform an operation.
This chapter outlines basic procedures that will get you started in Logic Developer - PLC to work with PACMotion.

**Location in the Navigator**

A target’s PACMotion node appears in the Project tab of the Navigator.

**To locate a target’s PACMotion node**

- Expand the target.

By default, the PACMotion node does not appear in the Navigator. You must add the PACMotion component.

**To add the PACMotion component to a PACSystems RX3i target**

- In the Project tab of the Navigator, right-click the target, point to Add Component, and choose PACMotion.

The PACMotion node is added to the target.

**Cam Editor**

The Cam editor is an accessory for PACSystems RX3i PACMotion programming that enables you to create, edit, and manage Cam profiles. Each Cam profile is a user-defined curve that specifies the response of a slave servo to a master position index. Cam profiles are referenced by name and grouped into the Cam Profile Library and the Active Profiles folder. Each active profile is intended for use on a specific PACMotion module. The hardware components are specified in the Hardware Configuration (HWC) of the parent target.

You can reuse a Cam profile by right-clicking it and choosing Copy Profile.

**Working with the Cam Editor**

You can adjust the curves of your Cam profile to suit the specific needs of your project. With the Cam editor, you create profiles by defining points on a master/slave position curve. Groups of adjoining points are allocated to sectors. Each sector is assigned a polynomial curve fit order (1,2,3,5) that specifies how the curve will be interpolated between points.
To create a Cam profile

1. In the Project tab of the Navigator, expand the PACMotion node.
2. Right-click Cam Profile Library and choose New Profile.
   A new Cam profile with a default name is added to your project.

Data Logging

To generate .dlog files

1. In logic, create an instance of the MC_DL_CONFIGURE PACMotion function block. To specify which parameters you want to log in the .dlog file, edit the properties of the variable (of the DATA_LOG_PARAM_CONFIG data type) that is assigned to the ParameterConfig input.
   When the instance executes, data logging is configured.
2. In logic, create an instance of the MC_DL_ACTIVATE PACMotion function block.
   When the Enable input to the instance is set to On, data logging occurs.
3. In logic, create an instance of the MC_DL_GET PACMotion function block.
When the instance executes, it writes the data logged to the *dlog file specified by the DataLogFile input.

**To add a Data Logging Window (DLW)**

1. In the Project tab of the Navigator, locate the Data Logging Windows node, right-click it, and choose **Create New**.

2. If you are online to the PACSystems RX3i, the Create dialog box appears. Select one of the following:
   - Controller: The DLW is a *dlog file stored on the PACSystems RX3i.
   - PC: The DLW is a *dlw or *csv file stored on your computer.

3. Do as follows, depending on which window appeared:

   **If you selected Controller**  
   The Controller File Explorer window appears and displays the *dlog files found on the PACSystems RX3i.

   **If you selected PC or are offline**  
   The Select DLW to Load dialog box appears.

4. (Optional.) Enter a new name for the DLW.

   **Want to know more?** In the online help index, look up "PACMotion Instructions and Function Blocks", "IC695PMM335: PACMotion Digital Motion Control Module", "Cam Profile Library (PACMotion)", "Enumerated Data Types: an Overview (PACSystems)", "Reference ID Variables (RIVs)", "Data Logging Window (PACMotion)" and "adding views to DLWs".
Motion Programming

Logic Developer - PLC* supports motion programming for the DSM324i and Motion Mate DSM314 motion control modules. High performance, easy-to-use, these multi-axis motion control modules are highly integrated with the PACSystems* RX3i and the Series 90*-30 Controller logic solving and communication functions.

Both the DSM324i and the DSM314 support 10 motion blocks, 40 subroutines, and a maximum total of 1000 motion program statements. Logic Developer - PLC, making motion programming possible, supports the following motion editors:

- Motion editor
- Local Logic editor
- Cam editor

This chapter outlines basic procedures that will get you started with Logic Developer - PLC to create motion programs with these three motion editors.

To add a Motion component to a target

- In the Project tab of the Navigator, right-click the target, point to Add Component, and then choose Motion.

A Motion Program node is added to your project. Included are empty Motion Blocks, Local Logic, CAM Profiles and CAM Blocks folders.
Motion Editor

MOTION EDITOR

Logic Developer - PLC* includes a Motion editor, which enables you to create Motion blocks for the DSM324i and the DSM314. This text-based editor is configurable as to its appearance and behavior. Comments and white space are not considered to be Motion block statements. The Motion block programming syntax is different from Local Logic syntax (see page 157).

To add a Motion Block

1. In the \[ 🎨 \] Project tab of the 🎨 Navigator, expand the 🎨 Motion Program.
2. Right-click 🎨 Motion Blocks and choose New.
   A new empty Motion block with a default name is added to your project.
3. Rename the block as desired.

To open a motion block for editing

1. In the 🎨 Project tab of the 🎨 Navigator, expand the 🎨 Motion Program.
2. Expand 🎨 Motion Blocks and double-click the 🎨 Motion Block you want to open.
   The block opens for editing in the Motion editor.
Working with the Motion Editor

To insert a command

1. In the Motion editor, right-click and choose **Insert Keyword**.
   
   A smart list appears showing all available motion commands.

2. Select the appropriate command from the smart list and press **ENTER**.
   
   The command is placed in the motion editor.

Want to know more? In the online help index, choose “Motion Editor”.

---

The Feedback Zone indicates that a connection has been established with the Controller.

The Companion automatically provides information on what you click.
## Motion Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCEL</td>
<td>The ACCEL statement sets the axis acceleration for subsequent moves and remains in effect in a given block unless changed. Note: If a move instruction is executed before ACCEL, the tag Acceleration is used.</td>
</tr>
<tr>
<td>BLOCK NUMBER</td>
<td>Block numbers can be used as the destination of JUMP commands. Block numbers must be unique and can be between 1 and 65535.</td>
</tr>
<tr>
<td>CAM</td>
<td>The CAM statement starts CAM motion and specifies exit conditions.</td>
</tr>
<tr>
<td>CAM-LOAD</td>
<td>CAM-LOAD loads a parameter register with the starting location for a CAM slave axis.</td>
</tr>
<tr>
<td>CAM-PHASE</td>
<td>CAM-PHASE sets the phase for CAM commands.</td>
</tr>
<tr>
<td>CALL</td>
<td>The CALL command executes another block as a subroutine.</td>
</tr>
<tr>
<td>CMOVE</td>
<td>The CMOVE command programs a continuous move using the specified position and acceleration mode.</td>
</tr>
<tr>
<td>DWELL</td>
<td>DWELL causes motion to cease for a specified time period before processing the next command.</td>
</tr>
<tr>
<td>ENDPROG</td>
<td>The ENDPROG statement terminates a Motion program definition.</td>
</tr>
<tr>
<td>ENDSUB</td>
<td>The ENDSUB statement terminates a Motion subroutine definition.</td>
</tr>
<tr>
<td>JUMP</td>
<td>Jump to a block number or a sync block within the current program or subroutine. The jump may be unconditional or conditional based on the status of a CTL bit.</td>
</tr>
<tr>
<td>LOAD</td>
<td>Initializes or changes a parameter data register with a 32-bit two-complement integer value.</td>
</tr>
<tr>
<td>PMOVE</td>
<td>The PMOVE command programs a positioning move using the specified position and acceleration mode.</td>
</tr>
<tr>
<td>PROGRAM</td>
<td>The PROGRAM statement is the first statement in a motion program. The program statement identifies the program number (valid range: 1 through 10) and the axis configuration. Program definitions cannot nest.</td>
</tr>
<tr>
<td>SUBROUTINE</td>
<td>The SUBROUTINE statement is the first statement in a motion subroutine. The subroutine statement identifies the subroutine number (valid range: 1 through 40) and the axis configuration.</td>
</tr>
<tr>
<td>SYNC BLOCK</td>
<td>A sync block is a special case of a block number. A sync block can be used only in multi-axis programs.</td>
</tr>
<tr>
<td>VELOC</td>
<td>Sets the process VELOCITY used by subsequent motion program move commands and remains in effect until changed by another VELOC statement.</td>
</tr>
</tbody>
</table>
LOCAL LOGIC

A Local Logic block runs synchronously with the Motion block, but is independent of the Controller’s CPU scan. This enables the DSM324i or DSM314 to interact much more quickly with motion I/O signals on its faceplate connectors than would be possible if the logic for the signals were handled in the _MAIN program running on the Controller.

Local Logic language uses free-form, text-based circuits and contains basic mathematical and logical constructs. The Local Logic syntax enables you to assign a variety of logic tasks to your motion programs while working in conjunction with Controller Logic programs and motion blocks to yield a flexible programming environment. Because it uses straightforward, understandable syntax, it is easy to gain proficiency with this editor.

The Local Logic programming language supports assignments, conditional statements, arithmetic, logical, and relational statements. Local Logic provides you with access to motion Controller data, parameters using a fixed set of variables, control bits, and status bits:

- Parameter data - accessible from Local Logic host Controller and motion blocks. The parameter data are similar to variables in a program.
- CTL bits - enable the Local Logic block or host Controller to signal the motion block to start an event.
- Motion block numbers - the current block number can be used within the Local Logic block or host Controller to make an action occur only during a specific motion programming section.

To create a Local Logic block

1. In the Project tab of the Navigator, expand the Motion Program.
2. Right-click Local Logic and choose New.
   
   A new Local Logic block with a default name is created.
3. Rename the block as required.

To open a Local Logic block for editing

- In the Project tab of the Navigator, expand the Motion Program and double-click the Local Logic block.

The Local Logic block opens for editing in the Local Logic editor.
Working with the Local Logic Editor

The Local Logic editor has its own distinctive syntax for constructing Local Logic blocks.

To insert a Local Logic command:

1. In the Local Logic editor, right-click and choose Insert Keyword. A smart list displays all available Local Logic commands.
2. Select the desired command in the smart list and press ENTER. The command is inserted.

Note: You can drag variables from the Local Logic Variable Table (LLVT) to the Local Logic editor.
Local Logic Variables

Local Logic is designed to complement a Controller’s logic and mathematical capabilities. Solving small Local Logic and mathematical sets requires a tight synchronization with the controlled motion.

Logic Developer - PLC* includes a table containing Local Logic variables, the Local Logic Variable Table (LLVT), that you can drag into your Local Logic blocks. As illustrated in the diagram below, the LLVT has several tabs that organize the variables by category:

1. In the Project tab of the Navigator, expand the Motion Program.
2. Right-click Local Logic and choose Local Logic Variable Table.
   The “Which LLVT do you want?” help topic appears.
3. Select Motion Mate DSM314 or DSM324i.
   The LLVT appears in the Infoviewer, displaying variables or data on each tab:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Group</th>
<th>Description</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual_Position_1</td>
<td>16 bits</td>
<td>Status variables:</td>
<td>Actual_Position (user-defined)</td>
<td>X</td>
</tr>
<tr>
<td>Actual_Velocity_1</td>
<td>16 bits</td>
<td>Status variables:</td>
<td>Actual_Velocity (user-defined)</td>
<td>X</td>
</tr>
<tr>
<td>Analog_Input_1</td>
<td>Analog</td>
<td>Feedback:</td>
<td>Analog Input 1 or 32000 + minus 10^3</td>
<td>X</td>
</tr>
<tr>
<td>Analog_Input_2</td>
<td>Analog</td>
<td>Feedback:</td>
<td>Analog Input 2 or 32000 + minus 10^3</td>
<td>X</td>
</tr>
<tr>
<td>Axis_Axis1</td>
<td>16 bits</td>
<td>Status variables:</td>
<td>Status axis 1</td>
<td>X</td>
</tr>
<tr>
<td>Axis_0</td>
<td>16 bits</td>
<td>Status variables:</td>
<td>Motion program block number</td>
<td></td>
</tr>
<tr>
<td>Commanded_Position</td>
<td>16 bits</td>
<td>Status variables:</td>
<td>Commanded_Position (user-defined)</td>
<td>X</td>
</tr>
<tr>
<td>Commanded_Velocity</td>
<td>16 bits</td>
<td>Status variables:</td>
<td>Commanded_Velocity (user-defined)</td>
<td>X</td>
</tr>
<tr>
<td>Digital_Output_1</td>
<td>Digital</td>
<td>Feedback:</td>
<td>Digital Output 1</td>
<td></td>
</tr>
<tr>
<td>Digital_Output_2</td>
<td>Digital</td>
<td>Feedback:</td>
<td>Digital Output 2</td>
<td></td>
</tr>
<tr>
<td>Drive_Enable_1</td>
<td>16 bits</td>
<td>Status variables:</td>
<td>Drive_Enable (user-defined)</td>
<td>X</td>
</tr>
<tr>
<td>Encoder_Code_1</td>
<td>16 bits</td>
<td>Status variables:</td>
<td>Encoder_Code (user-defined)</td>
<td></td>
</tr>
<tr>
<td>Feedback_1</td>
<td>Digital</td>
<td>Feedback:</td>
<td>Feedback_1 (user-defined)</td>
<td>X</td>
</tr>
<tr>
<td>Feedback_2</td>
<td>Digital</td>
<td>Feedback:</td>
<td>Feedback_2 (user-defined)</td>
<td>X</td>
</tr>
<tr>
<td>Feedback_Ramp_1</td>
<td>Digital</td>
<td>Feedback:</td>
<td>Feedback_Ramp_1 (user-defined)</td>
<td>X</td>
</tr>
<tr>
<td>Feedback_Ramp_2</td>
<td>Digital</td>
<td>Feedback:</td>
<td>Feedback_Ramp_2 (user-defined)</td>
<td>X</td>
</tr>
<tr>
<td>Feedback_Ramp_3</td>
<td>Digital</td>
<td>Feedback:</td>
<td>Feedback_Ramp_3 (user-defined)</td>
<td>X</td>
</tr>
<tr>
<td>Feedback_Ramp_4</td>
<td>Digital</td>
<td>Feedback:</td>
<td>Feedback_Ramp_4 (user-defined)</td>
<td>X</td>
</tr>
<tr>
<td>Feedback_Ramp_5</td>
<td>Digital</td>
<td>Feedback:</td>
<td>Feedback_Ramp_5 (user-defined)</td>
<td>X</td>
</tr>
<tr>
<td>Global</td>
<td>16 bits</td>
<td>Status variables:</td>
<td>Global data such as module status code</td>
<td></td>
</tr>
<tr>
<td>CTL bits</td>
<td>Digital</td>
<td>Status variables:</td>
<td>DSM general Control/Status bits</td>
<td>X</td>
</tr>
<tr>
<td>Parameter Registers</td>
<td>Digital</td>
<td>Status variables:</td>
<td>DSM parameter data</td>
<td>X</td>
</tr>
</tbody>
</table>

To view the LLVT:

1. In the Project tab of the Navigator, expand the Motion Program.
2. Right-click Local Logic and choose Local Logic Variable Table.
   The “Which LLVT do you want?” help topic appears.
3. Select Motion Mate DSM314 or DSM324i.
   The LLVT appears in the Infoviewer, displaying variables or data on each tab:

- **Axis 1**: Variables specific to axis 1
- **Axis 2**: Variables specific to axis 2
- **Axis 3**: Variables specific to axis 3
- **Axis 4**: Variables specific to axis 4
- **Global**: Global data such as module status code
- **CTL bits**: DSM general Control/Status bits
- **Parameter Registers**: DSM parameter data
Motion Programming

Local Logic

The table has six columns:

<table>
<thead>
<tr>
<th>Name</th>
<th>Contains the variable name that is to be used within a Local Logic block</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>The data type for this variable. For example, 32-bit means that this variable is a 32-bit variable.</td>
</tr>
<tr>
<td>Group</td>
<td>The group this variable is placed in. For example, Faceplate I/O means that this variable refers to a point on the module faceplate.</td>
</tr>
<tr>
<td>Description</td>
<td>This column contains a textual description of the variable. When you hover the mouse pointer over the description, a tool tip displays the description.</td>
</tr>
<tr>
<td>R</td>
<td>This column indicates if the variable can be read by a Local Logic block.</td>
</tr>
<tr>
<td>W</td>
<td>This column indicates if the variable can be written to by a Local Logic block.</td>
</tr>
</tbody>
</table>

To insert a Local Logic variable

1. In the Local Logic editor, right-click and choose Insert Variable.
   A smart list appears prompting you to choose a Local Logic variable name.
2. Select a variable in the list and press ENTER.
   The variable is inserted in your Local Logic.

Local Logic Commands and Operators

With Local Logic, you can execute basic logic and mathematical functions on the DSM324i or Motion Mate DSM314 module. Commands use upper case characters only and are case sensitive.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS</td>
<td>TRUE -(minus) &lt;= (less than or equal to)</td>
</tr>
<tr>
<td>BWAND</td>
<td>FALSE / (divide) &lt;&gt; (not equal)</td>
</tr>
<tr>
<td>BWOR</td>
<td>IF * (multiply)</td>
</tr>
<tr>
<td>BWXOR</td>
<td>THEN :=(assign)</td>
</tr>
<tr>
<td>BWNOT</td>
<td>END_IF &gt; (greater than)</td>
</tr>
<tr>
<td>ON</td>
<td>MOD &lt; (less than)</td>
</tr>
<tr>
<td>OFF</td>
<td>+ plus =&gt; (greater than or equal to)</td>
</tr>
</tbody>
</table>

Want to know more? In the online help index, choose “Local Logic: an Overview.”
CAM EDITOR

The Cam editor is an accessory for Logic Developer - PLC* motion programming that provides a means to create, edit, and manage electronic Cam profiles. Each Cam profile is a user-defined curve that specifies the response of a slave servo to a master position index. Cam profiles are referenced by name in the parent motion program and grouped into Cam blocks. Each block is intended for download to a specific motion module via its Controller. The hardware components are specified in the Hardware Configuration (HWC) of the parent target.

You can reuse a Cam profile by including it in multiple Cam blocks. To include a Cam profile in a Cam block, see “To add an alias to a Cam block”, on page 163.

To create a Cam block
1. In the Project tab of the Navigator, expand the Motion Program.
2. Right-click Cam blocks and choose New.
   A new Cam block with a default name is created.
3. Rename the block as desired.

To import Cam blocks
1. In the Project tab of the Navigator, expand the Motion Program.
2. Right-click Cam blocks and choose Import from File.
   The Open dialog box appears.
3. Browse to the Cam block you want to import (.csv or .txt file).
4. Click Open.
   The imported block appears in your project.

To open a Cam block for editing
1. In the Project tab of the Navigator, expand the Motion Program.
2. Expand Cam Blocks and double-click the Cam block you want to open.
   The block opens for editing in the Motion editor.
Working with the Cam Editor

You can adjust the curves of your Cam profile to suit the specific needs of your project. With the Cam editor you create profiles by defining points on a master/slave position curve. Groups of adjoining points are allocated to sectors. Each sector is assigned a polynomial curve fit order (1,2,3) that specifies how the curve will be interpolated between points.

To create a Cam profile

1. In the Project tab of the Navigator, expand the Motion Program.
2. Right-click Cam Profiles and choose New.

A new Cam profile with a default name is added to your project.
To configure a Cam profile

1. In the Project tab of the Navigator, expand the Motion Program.
2. Right-click Cam Profiles, right-click a Cam profile, and choose Properties.
   The Inspector displays the properties of the Cam profile.
3. Adjust the properties of the Cam profile in the Inspector to specify its type and boundary conditions.

To edit a Cam profile

1. In the Project tab of the Navigator, expand the Motion Program.
2. Right-click Cam Profiles and double-click a Cam profile.
   A graphical representation of your profile appears in the profile editor and a numeric representation appears in the profile table.
3. Insert and move points in the profile editor or table.
4. Group points into sectors in the profile table and assign curve fit order to each sector.

To add an alias to a Cam block

Right-click method

1. In the Project tab of the Navigator, expand the Motion Program and expand Cam blocks.
2. Right-click a Cam block, point to Add Alias to, and then choose a profile.
   Note: Aliases correspond to Cam profiles within Cam blocks. In order to create aliases for Cam blocks, you must have previously created Cam profiles.

Drag and drop method

1. In the Project tab of the Navigator, expand the Motion Program.
2. Expand Cam Profiles and expand Cam blocks.
3. Drag a Cam profile and drop it onto a Cam block.
   The Cam profile is added to the Cam block.

Want to know more? In the online help index, choose "Cam".
Motion Programming
CAM Editor
Index

A
Access Control Setting 90
Accumulators Configuring 120
Active Blocks folder 139
Active LD DLB 139
Adding
Carrier/Base to VersaMax Remote I/O 81
DSM324i 76
EGD Component 145
Ethernet Modules (PACSystems) 64
Fan Kits to PACSystems RX7i 64
GBC 73
GE Remote I/O Targets 80
Modules to a Carrier/Base 82
Motion Mate DSM314 76
Produced Exchanges 146
Remote I/O Scanner 79
Targets 34
Address Use report 53
Advanced Math Instructions
FBD 118
IL 125
LD 104
ST 133
Aliases 163
Aliasing Variables 24
Analog Start Address 120
Application Structure reports 53
Arrays 23
Assigning
Indirect References 91
Instance Data to Built-In Function Blocks (LD) 95
Length to LD Instructions 95
Operands to IL Instructions 122
Parameters beside FBD Instructions 111
Variables to Operands 96
Built-In Function Blocks Assigning Instance Data to (in LD) 95

B
Basic Hot CPU Redundancy 70
Basic Instructions
IL 125
Bit Operations
FBD 118
IL 125
LD 104
Bit References 25
Bitwise Operators
ST 133
Blocks
C 5, 86, 87, 135
CAM 85, 87, 161, 163
FBD 86, 108
Function Blocks 85, 86
IL 86
Interrupt 85, 86, 87
LD 86, 93
Local Logic 85, 87, 157
_MAIN 87, 88, 89
Motion 85, 87, 154
Parameterized 85, 86, 127
Reports 53
Scheduling 89
ST 86
Blue Arrows. See Docking Markers
Bookmarking Favorites 30
BOOL Variables
Changing the State 124, 132
Forcing the State 103, 117, 124, 133
Removing the Force 124, 133
Turning On or Off 103
Boolean Start Address 120
Build Tab 44, 47

C
C Blocks 5, 86, 87, 135
Importing 135
Setting Parameters 136
C Programs 5, 85, 86, 87, 137
Diagram 85
Scheduling 88
Setting Parameters 138
Cam 150
CAM Blocks 85, 87, 161, 163
Aliases 163
Creating 161
Importing 161
Cam Editor 5, 150, 153, 161
Working with the CAM Editor 150
CAM Profile Library 149, 150
CAM Profile report 53
CAM Profiles 87, 139, 140, 150
Adjusting the Curves 150
Configuring 163
Creating 151, 162
Editing 150, 163
Carrier/Base 82
Adding Modules 82
Adding to VersaMax Remote I/O 81
Catalog 62, 63, 64
Changing the State of BOOL Variables 124, 132
Checking
FBD Blocks 113
LD Blocks 96
Coils
Justification 26
LD 104
Index

COM 35
COM ports 38
Comments
   IL 122
   ST 129
Communication
   Configuring with Controllers 38
   Ethernet 38
   Modem Connection 38
   Serial Connection 38
Companion Help 21, 44
Using 27
Component
   Adding the PACMotion component 150
Compound Structure Variables 23
Configuring
   Accumulators 120
   CAM Profiles 151, 162, 163
   Communication with Controllers 38
   Communications with EGD Configuration Server 146
Controllers 61
CPUs 42, 62
DSM314 76, 77
DSM324i 77
Ethernet Communication 38
Ethernet Daughterboard 66
Execution Scheduling 89
Hardware 37, 55
Modem Communication 38
PACSystems RX7i 58
Redundancy Systems 72
Remote I/O 5
Scheduling 90
Secondary HWC 74
Serial Communication 38
Target to use EGD Configuration Server 146
Targets 35
VersaMax Remote I/O 80
Consumed Exchanges
Synchronizing 148
Consumers (EGD) 143
Contact Information 13, 14
Contacts
   LD 104, 125
Context-Sensitive Help 21, 27
Control 5, 18
Control Instructions
   FBD 118
   IL 125
   LD 104
Controllers 44
   Configuring 61
   Families 35, 55
   Family Property 35
   Starting 49
   Status 35
   Stopping 49
Conversion Instructions
   FBD 119
   IL 125
   LD 105
   ST 134
Conversion Reports 37
Converting Targets 36
Copying
   Entire LD Blocks as text 97
   Modules 66
   Sections of LD Blocks as text 98
   Text into the LD Editor 98
Counters
   FBD 118
   IL 126
   LD 105
CPU
   Configuring 42, 62
   Replacing 61
CPU Redundancy Over Genius 70
CPU684 41
CPU374 41
Creating
   CAM Blocks 161
   CAM Profiles 151, 162
   FBD Blocks 108
   IL Blocks 120
   LD Blocks 93
   Local Logic Blocks 157
   Parameterized ST Blocks 127
   Projects 18, 55, 80
   Reports 53
   ST Blocks 127
   User-Defined Folders 89
   User-Defined RVTs 52
   Variables 24
Variables from Names
   (IL) 122
   (ST) 129
Variables from Reference Addresses (IL) 122
CTL Bits (Local Logic) 157
Customer Care 13, 14
Customizing
   FBD Editor 108
   LD Editor 93
   ST Editor 127
D
Data Logging 151
Data Logging Window 149
Adding 152
Data Move Instructions
   FBD 119
   IL 126
   LD 105
   ST 134
Data Sources 152
Data Table Instructions
   IL 126
   LD 106
Data Values
   Monitoring (FBD) 117
   Monitoring (IL) 124
   Monitoring (LD) 99
   Monitoring (ST) 131
Data Watch Tool 20
Default RVTs 52
Diagnostics 50
DLB
   Active LD Diagnostic Logic Blocks 139
   Creating an active LD Diagnostic Logic Block 140
   Inactive LD DLBs 140
DLBs 5, 139
DLW
   Adding 152
   Docking Markers 21
   Documentation Address 35
Downloading IP Addresses non-PACSystems 42
PACSystems 40
Downloading to a Target 47
DSM314  76, 153, 154, 157, 159, 160  
Adding  76  
Configuring  76, 77  
DSM324i  76, 87, 88, 153, 154, 157, 159, 160  
Adding  76  
Configuring  77  
Dual Hardware Configuration  73, 75  
Duplicate Addresses  26  
Duplicating  
  FBD Logic  116  
  IL Logic  122  
  LD Logic  98  
  ST Logic  130  

E  
Editing  
  CAM Blocks  151, 162  
  CAM Profiles  163  
  FBD Blocks  109  
  IL Blocks  121  
  LD Blocks  93, 97  
  LD Logic  97  
  Local Logic Blocks  157  
  Motion Blocks  154  
  Projects  18  
  ST Blocks  128  
  Variables  23  
Editors  
  CAM  5, 150, 153, 161  
  FBD  5, 108  
  IL  5, 120  
  LD  5, 93  
  Local Logic  5, 153, 158  
  Motion  5, 153, 154  
  ST  5, 127  
EGD  
  Adding EGD Component  145  
  Adding Produced Exchanges  146  
  Configuration Server  143  
  Configuring Communications with EGD  
    Configuration Server  146  
  Configuring Target to use EGD Configuration Server  146  
Consumers  143  
Exchanges  143  
Installing EGD Configuration Server  145  
Installing EGD Management Tool  146  
Management Tool  144  
Pages  143  
Producers  143  
Publishing Produced Exchanges to EGD Configuration Server  147  
Signatures  144  
Synchronizing Consumed Exchanges  148  
Unconsumed Data Report  144  
Variable Editor  147  
EGD Configuration Server  143  
  Configuring Communications  146  
  Installing  145  
  Publishing Produced Exchanges to  147  
EGD Management Tool  144  
  Installing  146  
EGD reports  53  
EGD Variable Editor  147  
Enabling I/O Variables for a Module  68  
Enumerated Data Types  149  
Ethernet  35, 37, 38, 64, 66  
  Adding Modules  64  
Ethernet Adapter  57  
Ethernet Daughterboard  
  Configuring  66  
Ethernet Global Data. See EGD Exchanges (EGD)  143  
Executing Programs  49  
Execution Scheduling  
  Configuring  89  
Expansion Racks  57, 58, 63  
Extentions  
  .csv  152  
  .dlog  151  
  .dlw  152  
F  
Families  35, 55  
Fan Kits  
  Adding to PACSystems RX7i  64  
  Non-Intelligent  57  
Fault Locating References  37  
Fault Table Reports  50  
Fault Tables  50  
Favorites  30  
FBD  
  Drawing Wires  114  
  Negating Parameters  115  
  Negating Wires  115  
FBD Blocks  86, 108  
  Checking  113  
  Creating  108  
  Editing  109  
  Validating  113  
FBD Editor  5, 108  
  Customizing  108  
  Offline  109  
  Online  117  
FBD Instructions  118  
  Advanced Math  118  
  Assigning Parameters beside  111  
  Bit Operations  118  
  Changing the Number of Inputs  114  
  Control  118  
  Conversion  119  
  Counters  118  
  Data Move  119  
  Inserting  111  
  Math  119  
  Program Flow  119  
  Relational  118  
  Timer Function Blocks  119  
FBD Logic  108  
  Duplicating  116  
  Moving  116  
Feedback Zone  20, 44, 47, 53, 54  
Files  
  .csv File  152  
  .dlog File  151  
  .dlw File  152  
Finding text  
  Global Search  91  
  In One Block  90  
Folders
Index

Importing 19
Forces in Controller report 53
Forcing BOOL Variables 103, 117, 124, 133
Full-text Search
  Bookmarking Favorites 30
  Using 29
Function Block Instances 85, 86
Function Blocks 85
  Built-In 95
  Specialty 86
  User-Defined 86
fxClasses 20
G
  GBC 74, 78
    Adding 73
  GE Remote I/O
    Adding 80
  Generating Reports 53
  Genius Bus Controller. See GBC
  Genius Devices 58
  Genius Redundancy 70, 71
  Global Search 91
  GlobalCare 13, 14
  Going Offline from a Target 46
  Going Online to a Target 45
  GSDML files 141
H
  Hardware
    Configuring 37, 55
    Hardware Configuration 5, 37, 55
      Dual 75
      New Reference Assignment 26
      Primary 70, 75
      Secondary 70, 74, 75
      Synchronizing 75, 77
    Hardware Configuration report 53
    Hardware Key Authorization 8
  Help
    Companion 27
    Context-Sensitive 21, 27
    Favorites 30
    Full-text Search 29
    Index 28
    InfoViewer 27
    Table of Contents 31
Hot CPU Redundancy
  Setting Up Primary HWC 72
Hot CPU Redundancy Over
  Genius 71
  Hot Redundancy
    Configuring Systems 72
    CPU Over Genius 71
  Hot Redundancy Systems 70
  HWC. See Hardware Configuration
  Hyperlinks 54
I
  I/O Variables 22, 67
  IC695CPU320 61
  IC695PSA040 61
  IL Block Comments
    Inserting 122
  IL Block report 53
  IL Blocks 86
    Creating 120
    Editing 121
  IL Editor 5, 120
    Offline 121
    Online 123
  IL Inline Comments
    Inserting 122
  IL Instructions 125
    Advanced Math 125
    Basic Instructions 125
    Bit Operations 125
    Control 125
    Conversion 125
    Counters 126
    Data Move 126
    Data Table 126
    Inserting 121
  IL Logic 120
    Duplicating 122
    Moving 122
    Reformattting 123
  Importing
    C Blocks 135
    CAM Blocks 161
    Folders 19
  Inactive Blocks folder 139
  Inactive LD Diagnostic Logic Block 139
  Inactive LD DLBs 140
  Index 28
  Indirect References 91
  Assigning 91
  InfoViewer 21, 53, 54
  InfoViewer Help 27
  Initial Force States in Project report 53
  Inserting
    FBD Instructions 111
    IL Block Comments 122
    IL Inline Comments 122
    IL Instructions 121
    LD Instructions 95
    ST Block Comments 129
    ST Instructions 129
    ST Keywords 129
    ST Line Comments 129
    Variables (ST) 129
  Inspector 20
  Installation 7
  Installing
    EGD Configuration Server 145
    EGD Management Tool 146
  Instance Data
    Assigning to Built-in Function Blocks (LD) 95
  Instances of Function Blocks 85, 86
  Instruction List (IL) 120
  Instructions
    FBD 118
    IL 125
    LD 103
    ST 133
  Interacting with Controllers 44
  Internet Explorer 21
  Interrupt Blocks 85, 86, 87
  IP Address 35, 42
  Downloading to non-PACSystems 42
  Downloading to PACSystems 40
  Setting for Lower-End Series 90-30 42
  Setting for PACSystems (Permanent) 40
  Setting for Series 90-30
CPU364 41
Setting for Series 90-30
CPU374 41
Setting for Series 90-70 40
Setting Temporary 39

K
Keywords
Inserting (ST) 129

L
LD Block report 53
LD Blocks 86, 93
Checking 96
Copying Entire as Text 97
Copying Sections as Text 98
Creating 93
Editing 93, 97
Printing 54
Validating 96
LD Editor 5, 93, 97
Copying Text into 98
Customizing 93
Offline 94
Online 99
LD Instructions 103
Advanced Math 104
Bit Operations 104
C0ils 104
Contacts 104, 125
Control 104
Conversion 105
Counters 105
Data Move 105
Data Table 106
Inserting 95
Math 106
Program Flow 107
Relational 107
Timer Function Blocks 107
LD Logic 93, 97
Duplicating 98
Moving 98
Length
Assigning to LD Instructions 95
LLVT 159
Local Logic 153, 157
Local Logic Block report 53

CPU364 41
Setting for Series 90-30
CPU374 41
Setting for Series 90-70 40
Setting Temporary 39

K
Keywords
Inserting (ST) 129

L
LD Block report 53
LD Blocks 86, 93
Checking 96
Copying Entire as Text 97
Copying Sections as Text 98
Creating 93
Editing 93, 97
Printing 54
Validating 96
LD Editor 5, 93, 97
Copying Text into 98
Customizing 93
Offline 94
Online 99
LD Instructions 103
Advanced Math 104
Bit Operations 104
C0ils 104
Contacts 104, 125
Control 104
Conversion 105
Counters 105
Data Move 105
Data Table 106
Inserting 95
Math 106
Program Flow 107
Relational 107
Timer Function Blocks 107
LD Logic 93, 97
Duplicating 98
Moving 98
Length
Assigning to LD Instructions 95
LLVT 159
Local Logic 153, 157
Local Logic Block report 53

CPU364 41
Setting for Series 90-30
CPU374 41
Setting for Series 90-70 40
Setting Temporary 39

K
Keywords
Inserting (ST) 129

L
LD Block report 53
LD Blocks 86, 93
Checking 96
Copying Entire as Text 97
Copying Sections as Text 98
Creating 93
Editing 93, 97
Printing 54
Validating 96
LD Editor 5, 93, 97
Copying Text into 98
Customizing 93
Offline 94
Online 99
LD Instructions 103
Advanced Math 104
Bit Operations 104
C0ils 104
Contacts 104, 125
Control 104
Conversion 105
Counters 105
Data Move 105
Data Table 106
Inserting 95
Math 106
Program Flow 107
Relational 107
Timer Function Blocks 107
LD Logic 93, 97
Duplicating 98
Moving 98
Length
Assigning to LD Instructions 95
LLVT 159
Local Logic 153, 157
Local Logic Block report 53

CPU364 41
 Setting for Series 90-30
CPU374 41
 Setting for Series 90-70 40
Setting Temporary 39

K
Keywords
Inserting (ST) 129

L
LD Block report 53
LD Blocks 86, 93
Checking 96
Copying Entire as Text 97
Copying Sections as Text 98
Creating 93
Editing 93, 97
Printing 54
Validating 96
LD Editor 5, 93, 97
Copying Text into 98
Customizing 93
Offline 94
Online 99
LD Instructions 103
Advanced Math 104
Bit Operations 104
C0ils 104
Contacts 104, 125
Control 104
Conversion 105
Counters 105
Data Move 105
Data Table 106
Inserting 95
Math 106
Program Flow 107
Relational 107
Timer Function Blocks 107
LD Logic 93, 97
Duplicating 98
Moving 98
Length
Assigning to LD Instructions 95
LLVT 159
Local Logic 153, 157
Local Logic Block report 53

CPU364 41
 Setting for Series 90-30
CPU374 41
 Setting for Series 90-70 40
Setting Temporary 39

K
Keywords
Inserting (ST) 129

L
LD Block report 53
LD Blocks 86, 93
Checking 96
Copying Entire as Text 97
Copying Sections as Text 98
Creating 93
Editing 93, 97
Printing 54
Validating 96
LD Editor 5, 93, 97
Copying Text into 98
Customizing 93
Offline 94
Online 99
LD Instructions 103
Advanced Math 104
Bit Operations 104
C0ils 104
Contacts 104, 125
Control 104
Conversion 105
Counters 105
Data Move 105
Data Table 106
Inserting 95
Math 106
Program Flow 107
Relational 107
Timer Function Blocks 107
LD Logic 93, 97
Duplicating 98
Moving 98
Length
Assigning to LD Instructions 95
LLVT 159
Local Logic 153, 157
Local Logic Block report 53

LD Logic Blocks 85, 87
Creating 157
Editing 157
Local Logic Editor 5, 153, 158
Locating
 All Occurrences of a Variable 130
Logic
 FBD 108
 IL 120
 LD 93, 97
 Local 153
 ST 127
 Logic Developer - PC 5, 18
 Logic Developer - State 5, 18
 Logic Folder 86
 Logic Printouts 53
 Logic Programs and Blocks 85
 Logicmaster 5, 18

M
_MAIN Block 87, 88, 89
Main Program 88
Blocks 86
Diagram 85
Downloading 87
Number of Blocks 88
Properties 89
Making Word-for-Word Changes 102
Mapping a Variable to a Terminal 68
Math Instructions
 FBD 119
 IL 126
 LD 106
 ST 134
 Maximum Number of Blocks 88
 Modbus Address report 53
 Modern 35
 Modern Communication 38
 Module Catalog 41, 65, 81, 82, 83
 Modules
 Copying 66
 Moving 67
 Monitor Mode 44
 Monitoring
 Values of Parameterized Block Parameters (ST) 132
 Monitoring Data Values

N
Navigator 20, 52, 86, 150
Negating
 FBD Parameters 115
 FBD Wires 115
 Network Administrator 41
 Network Interface Unit (NIU) 80
 New Reference Assignment 26
 NIU 80
 Number of Blocks in the Main Program 88

O
Offline 44
 FBD Editor 109
 Going Offline from a Target 46
 IL Editor 121
 LD Editor 94
 ST Editor 128
 Online
 FBD Editor 117
 Going Online to a Target 45
 IL Editor 123
 LD Editor 99
 ST Editor 131
 Online in Monitor Mode 44
 Online in Programmer Mode 44

GFK-1918R
Opening a Project 19
Operands
Assigning to IL Instructions 122
Assigning Variables 96
Options Tab 52

P
PACMotion 5, 149
Cam profiles 149
components 149
Enumerated Data Types 149
Module 150
Node 149, 150
Reference ID variables 149
PACMotion node
Location in the Navigator 150
PACSystems 88, 144
Downloading IP Addresses 40
LD Instructions Support 104
Setting Permanent IP Addresses 40
PACSystems RX3i 33, 150
Illustration 57
Motion modules 76
Motion Program 88
PACSystems RX7i 33, 102
Adding Fan Kits 64
Configuring 58
Illustration 59
PACSystems RXi 33
Pages (EGD) 143
Parameter Data (Local Logic) 157
Parameter Editor 42, 62, 77
Parameterized Block Parameters
Viewing Values (ST) 132
Parameterized Blocks 85, 86
Creating (ST) 127
Parameters
C Blocks 136
C Programs 138
Permanent IP Address
Setting for PACSystems 40
Power Supply 80
Replacing 61
Primary HWC 70, 75
Setting Up for Hot CPU
Redundancy 72
Printing

LD Blocks 54
Reports 54
Produced Exchanges
Adding 146
Producers (EGD) 143
Product Authorization 8
Proficy Logic Developer - PC 5, 18
Proficy Logic Developer - State 5, 18
Proficy Motion 5, 18
Proficy View 5, 18
PROFINET 141
Program Blocks Folder 86
Program Flow Instructions
FBD 119
IL 126
LD 107
Programmer Mode 44
Programs 138
C. See C Programs
Comparative Diagram 85
Main. See Main Program
Motion. See Motion Program 85
Scheduling 88, 89
Types 85
Programs and Blocks 85
Project Tab 86
Projects 15
Creating 18, 55, 80
Downloading 47
Editing 18
Executing 49, 160
Importing Folders 19
Opening 19
Uploading 47
Published Variable Table 140
Publishing
Target's Produced Exchanges to
EGD Configuration 147
PVT
Published Variable Table 140

Q
Quick Start 16

R
Racks 55, 80
Expansion 57, 58, 63
Replacing 63
Redisplaying Reports 54
Redundancy
Basic Hot CPU 70
CPU Over Genius 70
Genius 70, 71
PROFINET 141
Wizards 72, 73
Reference Address 22, 52, 95
Reference ID variables 149
Reference View Tables 52
Default 52
User-Defined 52
Working with 53
Reformatting IL Logic 123
Relational Instructions
FBD 118
IL 126
LD 107
Relational Operators
ST 134
Remote I/O 34, 55, 81
Configuring 5
Remote I/O Scanner 78
Adding 79
Removing the Force from BOOL
Variables 124, 133
Replacing
CPUs 61
Power Supplies 61
Racks 63
Text 90
Reports 53
Address Use 53
Application Structure 53
CAM Profile 53
Conversion 37
EGD 53
Fault Table 50
Forces in Controller 53
Generating 53
Hardware Configuration 53
IL Block 53
Initial Force States in Project 53
LD block 53
Local Logic Block 53
Modbus Address 53
Motion block 53
Printing 54
Index

Redisplaying 54
Structured Text Blocks 53
Variable Difference 53
Variables 53
Reports Tab 53, 54
Right-Click Menu
HWC 61
Target 44, 153
RIVs 149
Run Mode 49
Run Mode Store 99, 101, 103
Rungs (LD Logic) 93
Run-time Destinations 33
Run-time Files 47

S
SBA 78
Scheduling
Blocks 89
Configuring 90
Programs 88, 89
Searching
Global Search 91
In One Block 90
Secondary HWC 70, 75
Configuring 74
Selecting
Range of ST Logic 130
Serial Communication 38
Series 90 Micro 33, 88, 102
Series 90-30 33, 58, 76, 88, 102, 120
Motion Program 87
Series 90-30 Ethernet Network
Interface Unit 33
Series 90-70 33, 58, 88, 89, 102, 104, 137
C Programs 87
Remote I/O Scanner 33
Series 90-70 Genius Remote I/O Scanner 33, 78
Setting Access Control 90
Setting IP Addresses
Lower-End Series 90-30 42
PACSystems (Permanent) 40
Series 90-30 CPU364 41
Series 90-30 CPU374 41
Series 90-70 40
Temporary 39

Setting Parameters
C Blocks 136
C Programs 138
Setting Up
Primary HWC for Hot CPU Redundancy 72
Signatures (EGD) 144
Slots 57, 61
Software Key Authorization 9
Specialty Function Blocks 86
ST Blocks 86
Creating 127
Editing 128
ST Editor 5, 127
Customizing 127
Offline 128
Online 131
ST Function Blocks
Timer Function Blocks 134
ST Functions
Advanced Math 133
Bitwise Operators 133
Conversions 134
Data Move 134
Math 134
Relational Operators 134
ST Instructions
Inserting 129
ST Logic 127
Duplicating 130
Moving 130
Selecting Range of 130
ST Statements 133
Starting
Controller 49
Machine Edition 16
Statements (ST) 133
Status 35
Status Bar 45
Stop Mode 49
Stopping 49
Stopping a Controller 49
Structured Text Blocks report 53
Sweep Time 35
Symbolic Variables 22
Synchronizing
Consumed Exchanges (EGD) 148

HWCs 75, 77
System Requirements 6

T
Table of Contents 31
Target Properties 35
Targets 86
Adding 34
Adding GE Remote I/O 80
Configuring 35
Converting 36
Definition 33
Downloading 47
Going Offline 46
Going Online 45
Offline 44
Online in Monitor Mode 44
Online in Programmer Mode 44
Remote I/O Scanner 78
Right-Click Menu 44, 153
Run Mode 49
Stop Mode 49
Uploading 48
Validating 44
Technical Support 13, 14
Templates 18, 34, 80
Terminal 67
Test Edit (LD) 100
Text
Replacing 90
Timer Function Blocks
FBD 119
IL 126
LD 107
ST 134
Toolchest 20
Tools 20
Docking Markers 21
Transacted Online Programming (LD) 100

U
UDT 23
Unconsumed Data Report (EGD) 144
Uploading from a Target 48
Uploading from the Controller 139
URL 35
<table>
<thead>
<tr>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User-Defined Folders</strong></td>
</tr>
<tr>
<td>- Creating 89</td>
</tr>
<tr>
<td><strong>User-Defined Function Blocks</strong></td>
</tr>
<tr>
<td>- 86</td>
</tr>
<tr>
<td><strong>User-Defined RVTs</strong></td>
</tr>
<tr>
<td>- Creating 52</td>
</tr>
<tr>
<td><strong>User-defined User Type</strong></td>
</tr>
<tr>
<td>- 23</td>
</tr>
<tr>
<td><strong>Using Help</strong></td>
</tr>
<tr>
<td>- Companion Help 27</td>
</tr>
<tr>
<td>- Favorites 30</td>
</tr>
<tr>
<td>- Full-text Search 29</td>
</tr>
<tr>
<td>- Help Index 28</td>
</tr>
<tr>
<td>- InfoViewer Help 27</td>
</tr>
<tr>
<td>- Table of Contents 31</td>
</tr>
<tr>
<td><strong>Using Templates to Create Projects</strong></td>
</tr>
<tr>
<td>- 18</td>
</tr>
<tr>
<td><strong>V</strong></td>
</tr>
<tr>
<td><strong>Validating</strong></td>
</tr>
<tr>
<td>- FBD Blocks 113</td>
</tr>
<tr>
<td>- LD Blocks 96</td>
</tr>
<tr>
<td>- Targets 44</td>
</tr>
<tr>
<td><strong>Variable Difference report</strong></td>
</tr>
<tr>
<td>- 53</td>
</tr>
<tr>
<td><strong>Variables</strong></td>
</tr>
<tr>
<td>- 22, 37</td>
</tr>
<tr>
<td>- Aliasing 24</td>
</tr>
<tr>
<td>- Arrays 23</td>
</tr>
<tr>
<td>- Bit References 25</td>
</tr>
<tr>
<td>- Compound Structures 23</td>
</tr>
<tr>
<td>- Creating 24</td>
</tr>
<tr>
<td>- Creating from Names</td>
</tr>
<tr>
<td>- (IL) 122</td>
</tr>
<tr>
<td>- (ST) 129</td>
</tr>
<tr>
<td>- Creating from Reference Addresses (IL) 122</td>
</tr>
<tr>
<td>- Editing 23</td>
</tr>
<tr>
<td>- I/O 22, 67</td>
</tr>
<tr>
<td>- Inserting (ST) 129</td>
</tr>
<tr>
<td>- Locating All Occurrences of 130</td>
</tr>
<tr>
<td>- Mapping 24</td>
</tr>
<tr>
<td>- Symbolic 22</td>
</tr>
<tr>
<td>- Turning On or Off 117</td>
</tr>
<tr>
<td><strong>Variables Reports</strong></td>
</tr>
<tr>
<td>- 53</td>
</tr>
<tr>
<td><strong>VersaMax</strong></td>
</tr>
<tr>
<td>- 33, 102</td>
</tr>
<tr>
<td>- Adding Carrier/Base to Remote I/O 81</td>
</tr>
<tr>
<td>- Configuring Remote I/O 80</td>
</tr>
<tr>
<td>- VersaMax Controllers 88</td>
</tr>
<tr>
<td>- 120</td>
</tr>
<tr>
<td>- VersaMax Ethernet RIOs 33</td>
</tr>
<tr>
<td>- VersaMax Genius RIOs 33</td>
</tr>
</tbody>
</table>

**W**
- Windows Explorer 20
- Wires
- Drawing in FBD 114
- Negating 115
- Word-for-Word Changes 100
- Making 102
- Working with Reference View Tables 53